Atmospheric Temperature Profile Retrievals Using 54 and 118-GHz Spectral Observations from the Proteus Aircraft

by

Andrew Sanchez

B.S.E., Arizona State University (1999)

Submitted to the Department of Electrical Engineering and Computer Science in partial fulfillment of the requirements for the degree of

Master of Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2002

Author.....
Department of Electrical Engineering and Computer Science
February 10, 2003

Certified by.....
David H. Staelin
Professor of Electrical Engineering
Thesis Supervisor

Accepted by.....
Arthur C. Smith
Chairman, Departmental Committee on Graduate Students

Atmospheric Temperature Profile Retrievals Using 54 and 118-GHz Spectral Observations from the Proteus Aircraft

by

Andrew Sanchez

Submitted to the Department of Electrical Engineering and Computer Science
On February, 03, 2002 in partial fulfillment of the
requirements for the degree of
Master of Science in Electrical Engineering and Computer Science

Abstract

Temperature profile retrievals based on passive microwave spectral observations from an aircraft were made with a Linear Least Squares Estimator (LLSE). The National Polar-orbiting Observational Environmental Satellite System (NPOESS) Aircraft Satellite Testbed-Microwave (NAST-M) is an imaging passive spectrometer observing oxygen absorption bands at 50-57 GHz and $118.75 \pm 0.8 - 118.75 \pm 3.5$ GHz.

NAST-M can accurately measure brightness temperatures aboard Scaled Composite's Proteus Aircraft from an altitude of 16 km. Simulating the brightness temperatures NAST-M would observe required the use of the TIGR profile set of 1761 radiosondes from around the world. Using the entire set of simulated data, the RMS retrieval error averaged <2K for pressure levels from the surface to the aircraft.

Data collected for retrievals were measured during CLOUDIOP, WVIOP, and AFWEX deployments over the Southern Great Plains. The flights occurred during March 2000, October 2000, and December 2000 respectively. Six flights have been studied for naturally occurring phenomenon. Using radiosondes collected from the Atmospheric Radiation Measurement (ARM) Program, a gain and baseline calibration correction was implemented to improve the accuracy of temperature profile retrievals for a particular day. Sensitivity of the NAST-M data was approximately ± 0.5K and ± 0.8K for 54 and 118-GHz spectrometers respectively, with the exception of channel 1 for both systems. Additive RMS noise due to calibration drift, interference, or unknown causes was calculated to be ~0.1K and ~0.5K respectively.

Thesis Supervisor: David H. Staelin
Title: Professor of Electrical Engineering

Acknowledgements

I would like to start out by giving much thanks to my wife Roni D. She has stood by my side through life's hardships ever since we met in 1995. I want to thank her for taking good care of our family and me while living at MIT. I want to thank my son Gabriel for always making me smile through his learning experiences. I want to thank my twins Adrianna and Alexander for making me realize that we can survive through anything. I want to thank my mom for always helping out when I was in financial turmoil. Helping us transplant across the country was a big help. Also I will never forget how you always push me to keep going when I decide to procrastinate at times. Thanks mom! I also want to thank my wife's parents, Ron and Brenda Johnson, for advice in life. You make me see that situations are really better then they appear. I want to thank my little bro, Frank, for helping me move out of the apartment this summer. Same goes for you too Jose and also for pointing out that I am such a pack rat. I want to thank my Dad, my Uncle Sang Sop, Larry, and my Halmoni for always being there. Thanks.

I want to thank Prof. David H. Staelin for his guidance, support, and patience throughout the course of this research. He has been a very important resource for completing this thesis. Thank you. I appreciate the opportunity of upgrading NAST-M. I want to give thanks to Phil Rosenkranz for answering my questions regarding atmospheric modeling. I want to thank R. Vincent Leslie, Fred W. Chen, and Bill Blackwell for helping me with my questions on NAST-M and MATLAB. I want to thank Chuck Cho for my Linux support. I would like to thank Amy Mueller for chatting with me once in a while and helping me with MATLAB. I would like to thank Dave Foss and Maxine Samuels of RLE for helping me with software and for ordering numerous circuit components for me. Finally, I give thanks to Marilyn Pierce for her suggestions and support. She has helped me in so many ways and has made my stay at MIT an experience I will never forget.

This research has been funded by the National Consortium for Graduate Degrees for Minorities in Engineering and Science, Inc., the Department of Electrical Engineering and Computer Science, MIT, and the Research Laboratory of Electronics, MIT.

Contents

List of Figures

List of Tables

Chapter 1

1 Introduction

1.1 Motivation

Our Earth's atmosphere has been studied for over the past fifty years with passive microwave instruments [1] [2] [3] [4]. Applications for remote sensing may include studying the oceans, the inner layers of the Earth's surface, to the atmosphere. Knowing more about the atmosphere may help to make better forecasts of weather patterns.

Using a space platform such as a satellite, to study the atmosphere is possible on a global scale. The next best platform is the use of high altitude testbeds to profile the atmosphere. The National Polar-orbiting Observational Environmental Satellite System (NPOESS) Aircraft Satellite Testbed-Microwave is a co-registered passive spectrometer with oxygen absorption bands at 50-57 GHz and $118.75 \pm 0.8 - 118.75 \pm 3.5$ GHz. The aircraft testbed for NAST-M was the Proteus of Scaled Composites. The aircraft typically flies at an altitude of 16 km. Using the data collected from deployments CLOUDIOP, WVIOP, and AFWEX, temperature profile retrievals will be performed using a linear estimator. The assumption of clear air will be used for the atmosphere.

1.2 Thesis Outline

Chapter 2 summarizes the issues of modeling the atmosphere for microwave sounding. The fundamentals of remote sensing are discussed along with the derivation of the radiative transfer equation. The pertinent mathematics required for modeling such a system and the linear least squares estimator are also discussed.

Chapter 3 summarizes the instrumentation and calibration for the NAST-M system. The collection method of radiosonde data from the Atmospheric Radiation Measurement (ARM)

Program is discussed. NAST-M data will be compared to the ARM data for corrections in calibrated brightness temperatures from deployments CLOUDIOP, WVIOP, and AFWEX.

Chapter 4 describes how the radiative transfer equation is inverted to retrieve temperature profiles. The development of such an algorithm requires the use of a priori knowledge which can be simulated from the TIROS Operational Vertical Sounder (TOV) Initial Guess Retrieval (TIGR) profile set. The algorithm can then be applied to measured data.

Chapter 5 summarizes and displays the retrieved temperature profiles from the CLOUDIOP, WVIOP, and AFWEX deployments. All retrievals are over the Southern Great Plains area over land in Oklahoma and Kansan, USA.

Chapter 6 summarizes the results and concludes the performance of the retrieval algorithm. Future work opportunities and suggestions are also included.

Chapter 2

2 Fundamentals of Remote Sensing

2.1 Physics

Within the microwave region of the electromagnetic spectrum (3 GHz – 300 THz), the thermal radiation emitted from sources such as the Earth's surface and atmosphere is directly proportional to the temperature [5]. For example, information about the atmosphere such as temperature and water vapor content can be extracted via passive remote sensing, or the science of measuring radiation from a distant location [6]. Finding a method of obtaining the temperature and water vapor content will rely heavily on estimation.

When modeling the Earth's atmosphere, one equation has been historically used, the Radiative Transfer Equation (RTE). The purpose of the equation is to relate the radiation emitted to the temperature and water vapor content of that atmosphere as a function of altitude. This distribution in altitude is often called a profile. By modifying the RTE in a discrete fashion, a forward model can be created to compute the radiances that an instrument looking downward into the atmosphere would measure [7] [8].

One such instrument is called the total power radiometer. The antenna of the total power radiometer is designed in such a way that only frequencies of interest will be allowed to pass through to the sensor. Each band of frequencies will be assigned a channel. A different weighting function characterizes each channel. The weighting function characterizes for each channel the part of the atmosphere to which the radiance measurements respond. Based on the radiances an instrument would observe, an inverse routine or retrieval algorithm can estimate the temperature and water vapor content that produced these radiances. One drawback is that many different atmospheric profiles can produce the same radiances. So the inverse routine may be singular [9].

2.1.1 Radiative Transfer Equation

The physics of thermal radiation can be described in the radiative transfer equation (RTE). The RTE is derived from the optical properties of the atmosphere [10] [11]. The transmittance of the atmosphere for a particular frequency is determined by the absorption of radiation by the atmosphere. Two assumptions made in the derivation are:

1. Temperature and water vapor content are constant for a given pressure level
2. The atmosphere and surface act like a blackbody.

The RTE can be modified to calculate the radiances that an instrument looking down into the atmosphere would measure based on a specific antenna pattern and the location of observation. The transfer of radiation in an incrementally small piece of atmosphere between the surface and the instruments antenna is:

$$dI_f = dI_{emission} + dI_{extinction} \quad (2.1)$$

Lambert's law states:

$$dI_{extinction} = -\alpha_f \cdot I_f \cdot ds \quad (2.2)$$

It should be noted that we are neglecting the scattering coefficient. α_f is called the absorption coefficient. Kirchhoff's law states:

$$dI_{emission} = \alpha_f \cdot B_f \cdot ds \quad (2.3)$$

Figure 2-1 illustrates the geometry that is being modeled. The distance along ds can be approximated to $\sec\theta \cdot dz$.

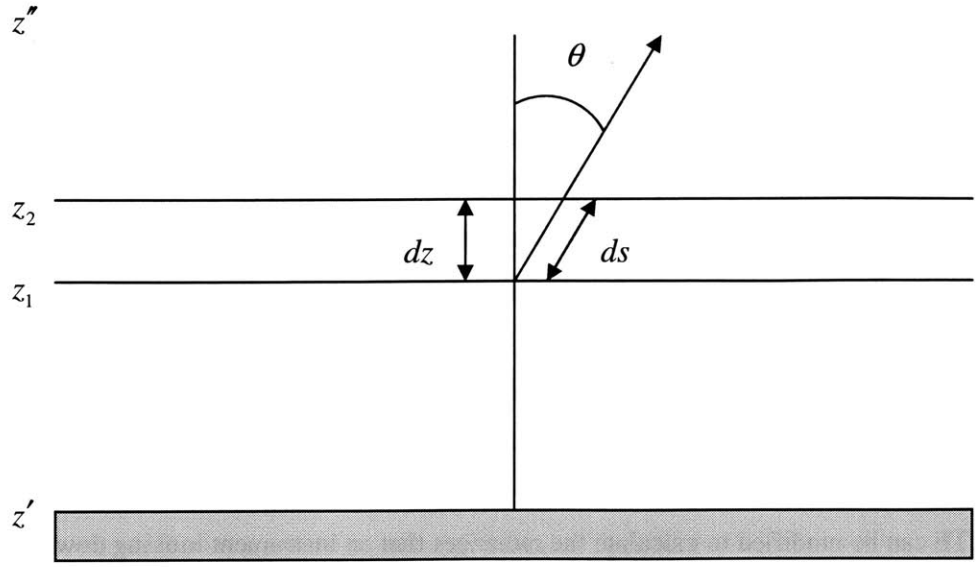


Figure 2-1 Geometric Layout for a Planar Stratified Atmosphere

The optical depth is:

$$\tau(z) = \int_{z}^{z''} \alpha_f(z) dz \qquad \textbf{(2.4)}$$

Substituting 2.2 and 2.3 into 2.1:

$$\frac{dI_f}{ds} = \alpha_f \left(B_f - I_f \right) \qquad \textbf{(2.5)}$$

Multiplying both sides by $e^{\tau(z)}$ and integrating from z' to z'':

$$I_f(z'') = I_f(z') \cdot e^{-[\tau(z') - \tau(z'')]\sec\theta} + \int_{z'}^{z''} \sec\theta \cdot B_f(z) \cdot e^{-[\tau(z) - \tau(z'')]\sec\theta} \cdot \alpha_f(z) \cdot dz \qquad \textbf{(2.6)}$$

Modifying equation 2.6 into brightness temperatures with $z' = 0$ and $\tau(z'') = 0$:

$$T_b = T_{b,surf} \cdot e^{-\tau(0)\sec\theta} + \int_0^{\infty} T(z) \cdot e^{-\tau(z)} \cdot \alpha_f(z) \cdot \sec\theta \cdot dz \qquad (2.7)$$

This form of the RTE only accounts for the thermal radiances from the surface and the atmosphere. Two additional sources need to be added. They are the downwelling and cosmic background radiation that reflects off the surface and to the instrument. The revise RTE is [12]:

$$T_b = T_{b,u} + \left[T_{b,surf} + (1-\varepsilon_{surf})T_{b,d}\right]E \qquad (2.8)$$

where:

$$T_{b,u} = \int_0^{\infty} T(z) \cdot e^{-\tau(z)} \cdot \alpha_f(z) \sec\theta \cdot dz \qquad (2.9)$$

$$T_{b,surf} = \varepsilon_{surf} \cdot T_{surf} \qquad (2.10)$$

$$T_{b,d} = T_{\cos mic} \cdot e^{-\tau(0)\sec\theta} + \sec\theta \int_0^{\infty} T(z) \cdot \alpha_f(z) \cdot e^{-[\tau(0)-\tau(z)]\sec\theta} dz \qquad (2.11)$$

$$E = e^{-\tau(0)\sec\theta} \qquad (2.12)$$

2.2 Mathematics

When modeling the clear air radiation observed over land by a downward looking instrument in the microwave region, most of the model can be considered linear. A linear least squares estimator (LLSE) can be implemented to perform temperature retrievals. There are other estimators such as a multilayer feedfoward neural network (MFNN) can also be implemented for retrievals. However the training of an MFNN can be a computational burden. For strictly temperature retrievals, the LLSE is the easiest and simplest solution.

2.2.1 Bayes' Least-Squares Estimation

Developing a Bayes' least-squares estimator would require the knowledge of a priori information about the random variable being estimated. The probability density function (PDF) of the desired random variable may also be known and can be used in the estimator.

2.2.2 Linear Least-Squares Estimator

The optimal estimator for estimating Y based on X with the MMSE criterion is:

$$\hat{Y}_{MMSE} = E[Y|X_1 = x_1, \mathrm{K}, X_L = x_L] = \int_{-\infty}^{\infty} y \cdot p_{Y|X}(y|X)dy \qquad (2.13)$$

X is a vector of given realization of random variables. Characterizing the complete statistical relationship between Y and X can be difficult. To make the estimator simpler (suboptimal), we will constrain it to an affine function of measured random variables:

$$\hat{Y}(X) = \alpha + \beta^T X \qquad (2.14)$$

The Minimum mean-square error (MMSE) criterion states:

$$\hat{Y}(X) = \arg\min_{\hat{a}(\bullet)} E\left[(Y - \hat{a}(X))^2\right] \qquad (2.15)$$

Minimizing the MMSE criterion will lead to finding a solution for the suboptimal estimator. One technique for minimizing this criterion is to let $\hat{a}(X) = \alpha + \beta^T X$ and taking partial derivative with respect to α and β. Set the derivatives to zero and solve simultaneously.

$$\frac{\partial}{\partial \alpha} E\left[\left(Y - \alpha - \beta^T X\right)^2\right] = -2E[Y - \alpha - \beta X] = 0 \qquad (2.16)$$

$$\frac{\partial}{\partial \beta} E\left[\left(Y-\alpha-\beta^T X\right)^2\right] = -2E\left[\left(Y-\alpha-\beta^T X\right)X\right] = 0 \tag{2.17}$$

Solving yields

$$\beta = \frac{E[YX]-E[Y]E[X]}{E\left[X^2\right]-E[X]^2} \tag{2.18}$$

$$\alpha = E[Y]-\beta E[X] \tag{2.19}$$

Define the covariance matrix of Y and X , covariance matrix of X :

$$K_{YX} = E[(Y-E[Y])(X-E[X])] = E[YX]-E[Y]E[X] \tag{2.20}$$

$$K_{XX} = E[(X-E[X])(X-E[X])] = E\left[X^2\right]-E[X]^2 \tag{2.21}$$

After solving the set for the coefficients, the suboptimal estimator is the linear least-squares estimator.

$$\hat{Y}(X) = E[Y]+K_{YX}\cdot K_{XX}^{-1}\cdot(X-E[X]) \tag{2.22}$$

Where K_{XX} is the covariance matrix of X and K_{YX} is the cross covariance matrix between Y and X . When the estimator is implemented using software, the sample covariance S_x^2 will be calculated to estimate the true covariance. Also the sample mean M_x will estimate the true mean.

$$S_x^2 = \frac{\sum_{i=1}^{N}(x_i-M_x)^2}{N} \tag{2.23}$$

$$M_x = \frac{\sum_{i=1}^{N} x_i}{N} \tag{2.24}$$

The sample cross covariance is given by:

$$S_{yx}^2 = \frac{\sum_{i=1}^{N} (y_i - M_y)(x_i - M_x)}{N} \tag{2.25}$$

2.2.3 Linear Regression

A linear relationship is defined by a slope and intercept. Data is often paired up as points such as (x_i, y_i). The statistical method of determining the slope and intercept parameters from a finite set of points is called linear regression. If the random variable X is independent, then it is called the regressor while the random variable Y, which is dependent on X, is called the response. X is assumed to be without error while Y is modeled as:

$$y_i = b + g \cdot x_i + e_i \tag{2.26}$$

e_i is considered the error from experimental data. If each point is independent and contain equal variance, then a linear system can be formed:

$$y = A \cdot \hat{\gamma} \tag{2.27}$$

where

$$y = \begin{bmatrix} y_1 \\ y_2 \\ \mathrm{M} \\ y_N \end{bmatrix} \qquad A = \begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ \mathrm{M} & \mathrm{M} \\ x_N & 1 \end{bmatrix} \qquad \hat{\gamma} = \begin{bmatrix} \hat{\alpha} \\ \hat{\beta} \end{bmatrix}$$

The linear system has the form:

$$A\hat{x} = b \tag{2.28}$$

With linear regression, the system is over determined and therefore can be solved in the same manner as the LLSE. By choosing the MMSE criterion, the linear regression solution is very similar to the LLSE. In this case,

$$\hat{x} = \left(A^T A\right)^{-1} A^T b \tag{2.29}$$

Therefore the solutions to the linear regression is

$$\hat{\beta} = \frac{N\sum_{i=1}^{N} y_i x_i - \left(\sum_{i=1}^{N} y_i\right)\left(\sum_{i=1}^{N} x_i\right)}{N\sum_{i=1}^{N} x_i^2 - \left(\sum_{i=1}^{N} x_i\right)^2} \tag{2.30}$$

$$\hat{\alpha} = M_y - \hat{\beta} M_x \tag{2.31}$$

Make note that the linear regression is the inferred form of the LLSE based on a finite set of points. More information of the mathematics can be found in [13] [14].

Chapter 3

3 Data

In preparation for developing a method to perform temperature retrievals, the necessary data to model an atmosphere will be required. Ultimately, the retrievals will be performed using NAST-M data. NAST-M is a passive microwave temperature sounder [15] [16] [17]. Currently, NAST-M is mounted underneath the fuselage of the Proteus aircraft. Before retrievals can be performed on data from the field, a set of simulated atmospheres will be needed to represent an ensemble of atmospheres to expect. The TIROS Operational Vertical Sounder (TOVS) Initial Guess Retrieval (TIGR) set will provide temperature, water vapor, and ozone profiles for many different climates and seasons. The TIGR set is compiled of 1761 profiles from the mid 70's to the late 80's. To provide truth to temperature profiles retrieved from NAST-M data, a set of radiosonde data from the Atmospheric Radiation Measurement (ARM) program will be collected for the areas flown over during the missions of CLOUDIOP, WVIOP, and AFWEX.

3.1 NAST-M Instrument Overview

NAST-M stands for NPOESS (National Polar-orbiting Observational Environmental Satellite System) Aircraft Satellite Testbed Microwave. Two total power radiometers near 54-GHz and 118-GHz in the oxygen absorption band are used to measure radiation in the atmosphere. The 54-GHz is a single sideband (SSB) super heterodyne receiver with a local oscillator (LO) frequency of 46 GHz with eight channels ranging from 50.2 GHz to 56.2 GHz. The 118-GHz is a double sideband (DSB) super heterodyne receiver that has nine channels of which the first six are useable. A list of the channels and frequencies can be found in Table 3.1 and Table 3.2. For each system, a weighting function is applied to each channel for the purpose of seeing a sounding from NAST-M. From an altitude of 20 km, the nadir spatial resolution is 2.6 km. NAST-M has an antenna reflector that scans $\pm 64.8^{\circ}$ from nadir and makes 19 measurements with spacing of 7.2°. For more information about NAST-M refer to [18] and [19].

The output from each receiver is a voltage value between ± 10 volts. The voltage value is then converted via A/D conversion to a value between $\pm 2^{16}$. After the conversion, it is saved to disk as a "count". There are 25 spots total for which a count is saved. Two measurements are made looking at the sky pipe (zenith), two are made looking at a hot load, one measurement is made for each of the 19 nadir spots, and two are made for the ambient load.

During the missions of CLOUDIOP, WVIOP, and AFWEX, NAST-M was mounted in a superpod underneath the Proteus' fuselage. The Proteus is a custom built aircraft from Scaled Composite LLC. Two Williams International FJ44-2E turbofan engines were selected for propulsion. Its unique shape allows for many different configurations (Figure 3-1) for mounting instruments for various testbeds. The Proteus has a payload of up to 2000 pounds and can cruise at altitudes above 18 km for a maximum of 14 hours. Two crucial pieces of data supplied by the Proteus are the static air pressure and the aircraft roll. The static pressure will be used to provide NAST-M's relative height above the earth's surface. The aircraft roll will be used to determine the flatness of a scan. Previously, the aircraft roll was used take the rolls out of calibrated data so that the brightness temperatures measured would seem as though the aircraft was flying perfectly flat. But because of the location, NAST-M's sky pipe would be blocked off by the fuselage of the Proteus. An attempt to mount NAST-M with a 10 degree tilt to point away from the fuselage was unsuccessful. The fuselage was still in the way. By introducing a 10 degree tilt, a spot shift of the nadir views from previous missions CAMEX and WINTEX was inevitable. Without the zenith view, the calibration routine for the previous missions would have to be modified so that only the hot load and the ambient load counts will be used in calibrating brightness temperatures.

Table 3.1 Spectral Description of the NAST-M 54-GHz System.

54-GHz System SSB LO = 46			
Ch	RF (GHz)	IF (MHz)	BW (MHz)
1	50.21 - 50.39	4210 - 4390	180
2	51.56 - 51.96	5560 - 5960	400
3	52.6 - 53	6600 - 7000	400
4	53.63 - 53.87	7630 - 7870	240
5	54.2 - 54.6	8200 - 8600	400
6	54.74 - 55.14	8740 - 9140	400
7	55.335 - 55.665	9335 - 9665	330
8	55.885 - 56.155	9885 - 10155	270

Table 3.2 Spectral Description of the NAST-M 118-GHz System.

118-GHz System DSB LO = 118.75				
Ch	LSB (GHz)	USB(GHz)	IF (MHz)	BW (MHz)
1	114.75 - 115.75	121.75 - 122.75	3000 - 4000	1000
2	115.95 - 116.45	121.05 - 121.55	2300 - 2800	500
3	116.45 - 116.95	120.55 - 121.05	1800 - 2300	500
4	116.95 - 117.35	120.15 - 120.55	1400 - 1800	400
5	117.35 - 117.75	119.75 - 120.15	1000 - 1400	400
6	117.75 - 118.15	119.35 - 119.75	600 - 1000	400
7	118.15 - 118.45	119.05 - 119.35	300 - 600	300
8	118.45 - 118.58	118.92 - 119.05	170 - 300	130
9	118.58 - 118.68	118.82 - 118.92	70 - 170	100

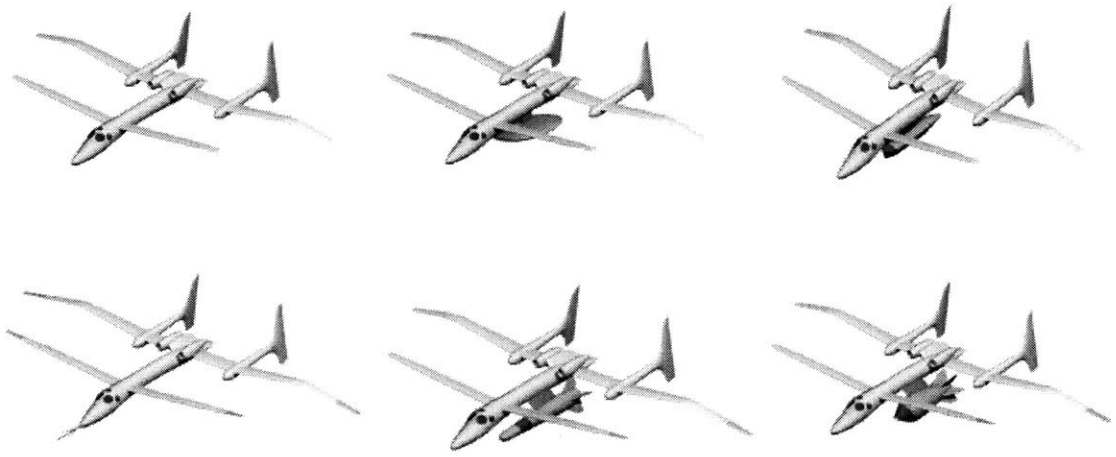


Figure 3-1 Different Configurations of the Proteus.

3.1.1 Calibration

During calibration, the raw data recorded by NAST-M (counts), is linearly transformed into brightness temperature in a least squares sense. Using the Proteus as an aircraft testbed resulted in losing the zenith view calibration load. Only the hot and ambient loads remain for calibration. The scan pattern is still the same starting with two measurements of the zenith view, two measurements of the hot load, one measurement for each of the 19 nadir views, and two

measurements of the ambient load. Each scan (all 25 spots) usually takes around 5.5 seconds to complete. During calibration, we simply ignore the zenith view counts. The calculated brightness temperature can be modeled with the following formula:

$$T_b = g \cdot C + b \tag{3.1}$$

The gain g and baseline b can be derived by solving a set of linear equations using the calibration points (C_H, T_H) and (C_A, T_A). The subscripts H and A represent the counts and measured temperature in Kelvin of the hot and ambient load respectively. During flight, the temperatures of both loads are measured by the Resistive Thermal Device (RTD) unit [19] within NAST-M. The solution for g and b is:

$$g = \frac{T_A - T_H}{C_A - C_H} \tag{3.2}$$

$$b = T_A - \frac{T_A - T_H}{C_A - C_H} C_A \tag{3.3}$$

This calculation is made for each spot. The estimated gain and baseline are then used to convert the counts for each spot into brightness temperature. This process repeats all over for the next scan.

3.1.2 Correction for thermal gradients in the hot load

Since the hot load is heated to maintain an average temperature of 334K at all altitudes (roughly up to 16 km), the hot load may not be uniformly 334K throughout the blackbody. The hot load has 7 RTD's measuring its temperature. To compensate for such a gradient, hot load weights for each of the RTD's have been calculated to find the best average temperature for the hot load.

3.1.3 Antenna beam spillover correction

Side lobes of an antenna pattern can play a significant role in the measurement of radiances. For example, when the antenna is pointed at the hot load, radiation contributions from the sky view and the nadir view will add to radiance measurement. This additional corruption will offset the actual brightness temperature of the hot load. The hot load will appear to be warmer. The same will happen for the ambient load. The corrupted temperature of the ambient load and hot load can be modeled as a linear combination of the spillover through the sky pipe, the nadir view, and the "true" load temperature as:

$$T_A' = \eta_Z^A T_Z + \eta_N^A T_N + \left(1 - \eta_Z^A - \eta_N^A\right) T_A \quad (3.4)$$

$$T_H' = \eta_Z^H T_Z + \eta_N^H T_N + \left(1 - \eta_Z^H - \eta_N^H\right) T_H \quad (3.5)$$

The four η values for each radiometer channel have been accurately measured in the lab. T_Z and T_N can be estimated from NAST-M data. With these values, the true load temperatures can be estimated along with the corruption corrected nadir brightness temperatures. More information can be found in appendix C of [19].

3.1.4 Instrument Sensitivity

Inherent to all systems is the presence of noise. The theoretical noise can be calculated from ΔT_{RMS}. ΔT_{RMS} can be calculated directly by knowing the receiver temperature, target temperature, bandwidth, integration time, and calibration error approximation. A different approach involves calibrating the counts and finding the standard deviation based on the calibrated brightness temperature on the calibration loads (Table 3.3). Actual measurements of the amount of standard deviation have been done in [18] and [19].

54-GHz System				118-GHz System			
Ch. 1	0.1879	Ch. 5	0.1248	Ch. 1	0.1922	Ch. 5	0.3002
Ch. 2	0.1274	Ch. 6	0.1528	Ch. 2	0.2436	Ch. 6	0.3814
Ch. 3	0.1084	Ch. 7	0.1754	Ch. 3	0.2066	Ch. 7	0.6080
Ch. 4	0.1474	Ch. 8	0.2321	Ch. 4	0.2679	Ch. 8	0.8930
						Ch. 9	1.1545

Table 3.3 Sensitivities for the NAST-M Instrument.

In training an estimator for temperature retrievals, the measurements can be used to simulate noise NAST-M would measure and added to simulated brightness temperatures.

3.1.5 Brightness Temperature Post Processing

In preparation of measured NAST-M data, two steps are made. Using Proteus navigational data, the aircraft roll is attempted to be removed. Anytime the aircraft roll more than 2 degrees, the brightness temperature value before and after the roll are interpolated to estimated the brightness temperature for that scan as if there were no roll.

The second step is to lessen the effects of noise. Smoothing of the data is performed with a triangular filter. This is accomplished by taking a boxcar and filtering the data for each spot, and for each channel. This result is then time reversed and filtered by the same boxcar. The final output is then time reversed to end up with filtered zero phased-shift brightness temperatures [20]. The boxcar filter is 4 samples long. The height of each sample is one over the length of the filter, in this case ¼.

3.2 TIGR profile set

The TIGR profile consists of a collection of 1761 temperature profiles [21]. Table 3.4 shows the distribution across climates. Each profile has been read into MATLAB using a script [appendix]. One profile contains 40 temperature measurements at 40 different pressure levels. Water vapor content has been calculated using [9]. Using these two measurements along with surface parameters, one can simulate brightness temperatures. Since NAST-M was flown at a mean pressure of 112 mbar during CLOUDIOP, WVIOP, and AFWEX, only the first 22 pressure levels will be attempted for retrieval. Each profile contains a surface temperature but it is only the surface air temperature. Since the retrievals will be done over land, a Gaussian random variable of zero mean and standard deviation of 3K will be added.

Climate Type	Number of Radiosondes
Tropical Atmosphere	322
Mid Latitude-1 Atmosphere	388
Mid Latitude-2 Atmosphere	354
Polar-1 Atmosphere	104
Polar-2 Atmosphere	593

Table 3.4 Distribution of TIGR Profiles Throughout the Climates.

3.2.1 Simulating Brightness Temperature Components

Since training of an estimator will be performed using the TIGR set, a pre-computed set of upwelling $T_{b,u}$, downwelling $T_{b,d}$, and one way transmittance E have been calculated and saved for future use. A MATLAB script to calculate these components can be found in the appendix. Anytime a set of brightness temperatures are needed for training or testing the performance of the estimator, a random set of surface emissivities, surface temperatures, and pressure levels of the instruments location above the earth can be generated to create a new realization of simulated brightness temperatures $T_{b,sim}$.

3.3 Atmospheric Radiation Measurement Program

During each mission, a set of radiosondes and surface air temperature measurements have been collected for the days NAST-M was flown. This small ensemble of radiosondes can be obtained from the Atmospheric Radiation Measurement Program (ARM). The ARM program is setup to measure the many uncertainties of global climate change. The area in particular is the Southern Great Plains which is mainly located in Oklahoma and Kansas, USA. Figure 3-2 is an example of a typical NAST-M trajectory of the Southern Great Plains. The set of ARM data will serve as a means of truth so that retrieved data can be compared. ARM will also serve as a method of calibration for NAST-M data.

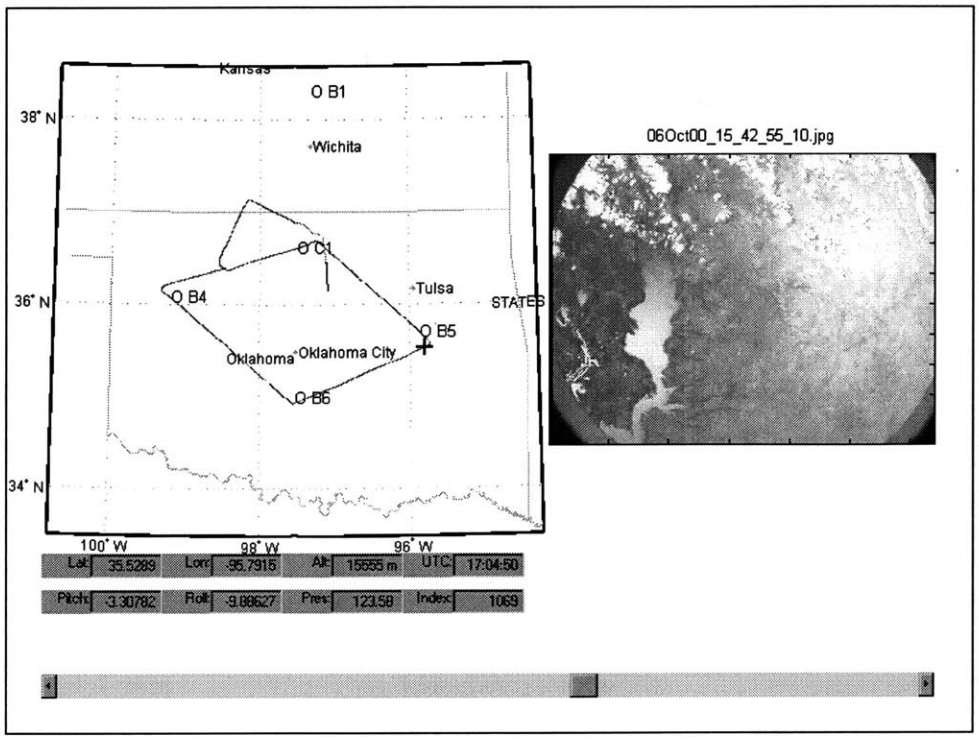


Figure 3-2 Southern Great Plains and Radiosonde Sites.

3.3.1 Partial atmosphere vs. Entire Atmosphere

The collection of radiosondes to be used to characterize the atmosphere NAST-M flew over, typically floated up to around the 22nd pressure level of the TIGR set. In the TIGR profile set, there are 40 pressure levels that characterize an atmosphere. When computing $T_{b,u}$, $T_{b,d}$, and E, ideally one would like to use the entire Earth's atmosphere. The TIGR set captures most of the atmosphere to be modeled. To characterize the atmosphere using an ARM radiosonde for a particular day and to compare it to the TIGR set would require using only the first 22 pressure levels. The 22 pressure levels in the TIGR set and the collection of radiosondes are satisfactory because NAST-M was flown around the same pressure level. The RMS errors resulting from using a partial atmosphere and a full atmosphere is shown in Figure 3-3.

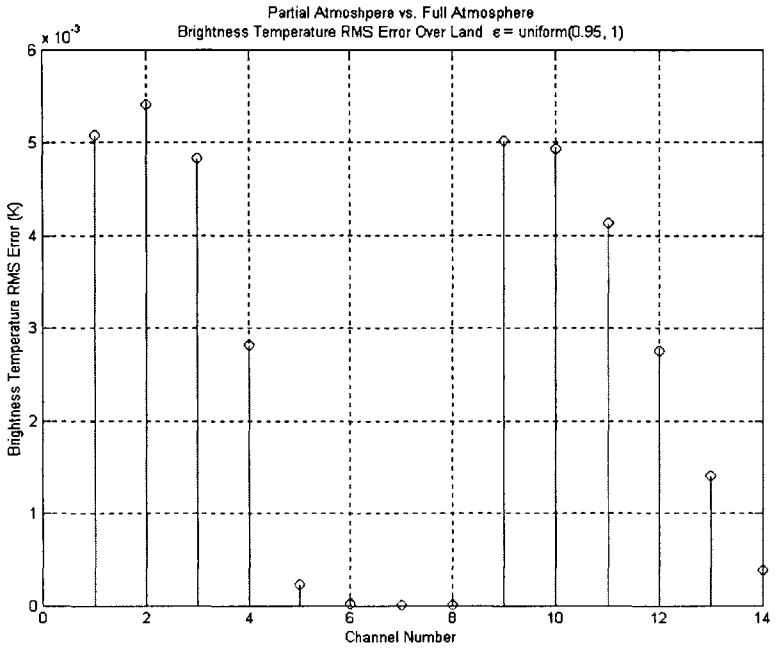


Figure 3-3 Partial Atmosphere vs. Full Atmosphere.

3.3.2 Simulation of Brightness Temperatures from ARM

When collecting usable ARM radiosondes, certain conditions must be met.

1. Based on the trajectory NAST-M was flown, a radiosonde launch site must be within in 50 km.
2. The radiosonde must have 22 pressure levels that match up with the TIGR pressure levels.
3. The duration of the radiosonde flight must have some overlap with the duration of NAST-M's flight.

It has been found that most of the ARM radiosondes collected begin floating from the surface to around the 22nd pressure level of the TIGR profile set. This works out great because the mean pressure NAST-M was flown was also around the 22nd pressure level of the TIGR set. With this small ensemble of radiosondes, brightness temperature components can be calculated. Using a partial atmosphere as opposed to a full atmosphere introduces very little error in calculating the brightness temperature components. Having calculated the brightness temperature components

from the radiosondes in the same manner as the TIGR set, simulated brightness temperatures can be randomly generated by introducing a random set of surface parameters

3.4 CLOUDIOP, WVIOP, and AFWEX Deployments

Missions CLOUDIOP, WVIOP, and AFWEX occurred over the southern Great Plains area (OK and KS). In attempts to retrieve temperature for these missions, the assumption of no liquid content in the atmosphere will be used. Historically for this region, land surface emissivities have ranged from 0.95 to 1. Also the surface emissivity of the 54-GHz is usually lower than the 118-GHz. To help model the surface emissivity of the 54-GHz, an emissivity value uniformly distributed between 0.95 and 1 will be generated. A zero-mean Gaussian with standard deviation 0.01 will be added to add a little randomness. The 118-GHz surface emissivity is found by using the following formula:

$$\varepsilon_{118} = \frac{1}{3} + \frac{2}{3}\varepsilon_{54} \tag{3.6}$$

A zero-mean Gaussian with standard deviation of 0.01 is added to the emissivity of the 118-GHz. This is important because during the training of an estimator, we don't want it to train to noise. Of course the randomly generated emissivities will have to be clipped to 1 if they exceed this value. The surface temperatures are generated by taking the first entry of each TIGR profile and adding a zero mean Gaussian random variable with a standard deviation of 3 K.

3.5 Bias removal

Throughout the flight of NAST-M, data will be measured at various pressure levels. When NAST-M is at the nominal pressure level, the brightness temperatures that have been calibrated from the raw counts will be similar to the simulated brightness temperatures from the ARM radiosondes. Brightness temperatures simulated from ARM radiosondes will be simulated from the same pressure level as NAST-M. This step will reveal that NAST-M brightness temperatures are slightly different than ARM brightness temperatures. Since ARM data is considered to be "truth", a small gain and baseline adjust will have to be made. For a particular day, an average of

different nadir view brightness temperatures near the launch site of the ARM radiosonde will be chosen to represent the same atmosphere the ARM radiosonde floated through. ARM brightness temperatures will be simulated from the same pressure level as NAST-M for each view. The difference will then be calculated:

$$\Delta T_{b,AN} = T_{b,ARM} - T_{b,NASTM} \tag{3.7}$$

Having these differences, a scatter plot can be made. For each measured brightness temperature, the associated difference will be plotted i.e. the difference in brightness temperature is a function the measured NAST-M brightness temperature. The differences for each measured brightness temperature will be different and therefore a single line can't be calculated deterministically to pass through all the points. The best solution is to find a gain g and baseline b of a linear regression that fits the data the best. Figure 3-4 is an example of finding the best fit.

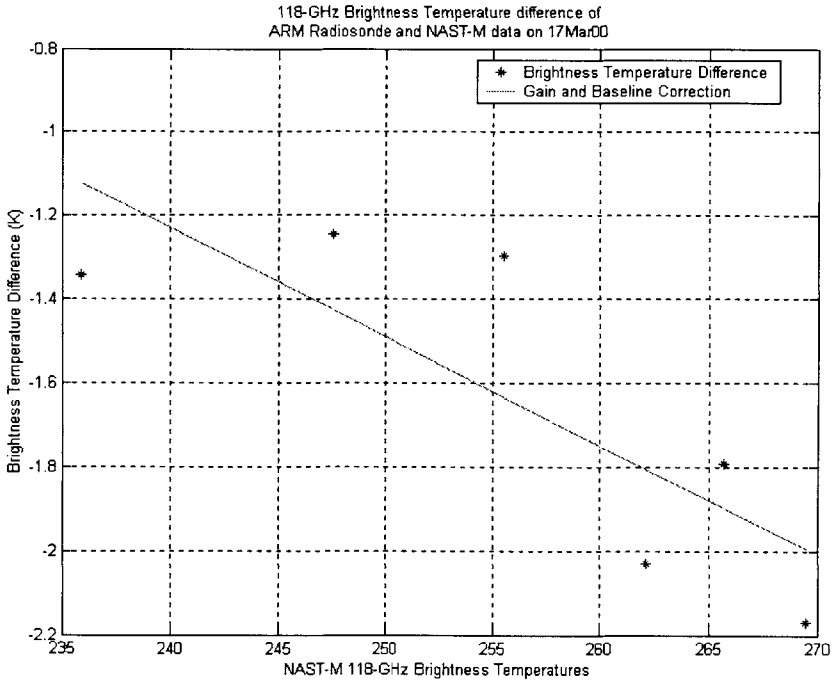


Figure 3-4 Example of Gain and Baseline Correction.

The same process is to be completed for the other days. Having calculated the gain and baseline for each radiosonde location, performing an average will help reduce accidental spatial or

temporal differences between the flight portion and weather related anomalies of a radiosonde. The corrected NAST-M brightness temperature is as follows:

$$T'_{b,NASTM} = T_{b,NASTM} + \Delta T'_{b,AN} \quad (3.8)$$

where:

$$\Delta T'_{b,AN} = g \cdot T_{b,NASTM} + b \quad (3.9)$$

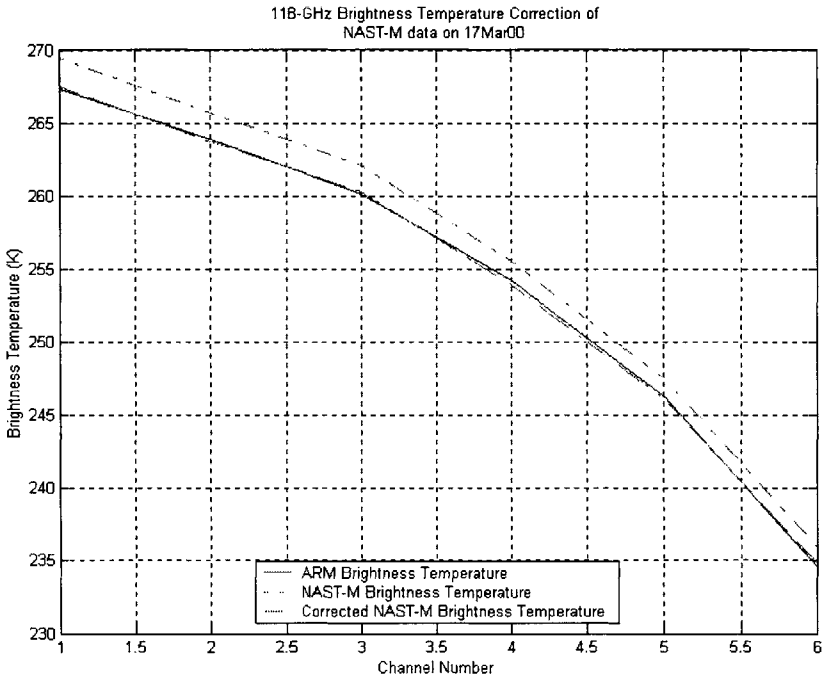


Figure 3-5 Example of Brightness Temperature Correction.

The gain and baseline calculations were found from the nadir view brightness temperatures. It can be shown that these calculations can be used for all angles. To show this, NAST-M brightness temperatures have been simulated for all spots. Then a comparison between measured and simulated data was plotted. Figure 3-6 shows 54-GHz measured data from March 17, 00. Figure 3-7 is the plot of simulated data. The comparison is shown in Figure 3-10.

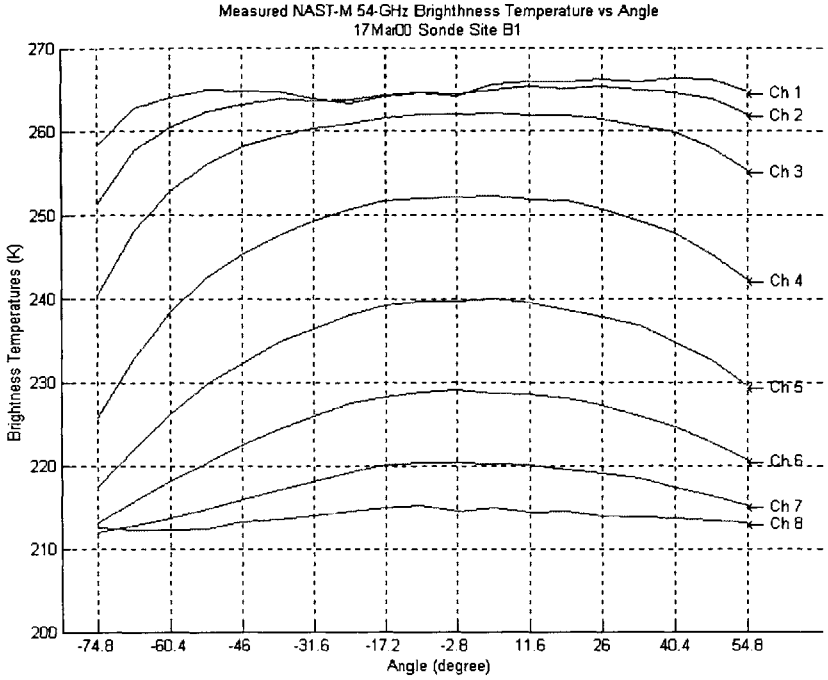


Figure 3-6 Measured 54-GHz data over all spots.

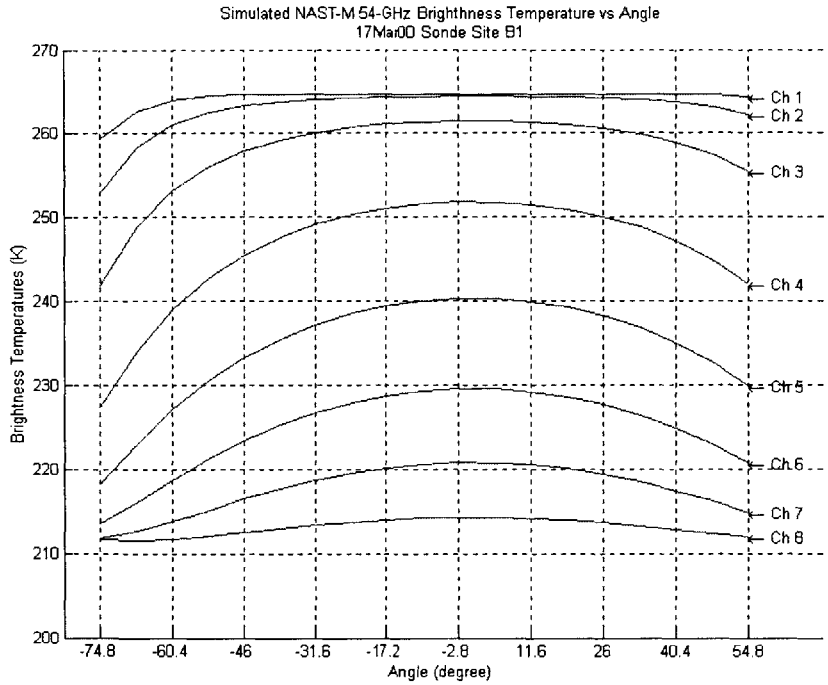


Figure 3-7 Simulated 54-GHz data over all spots.

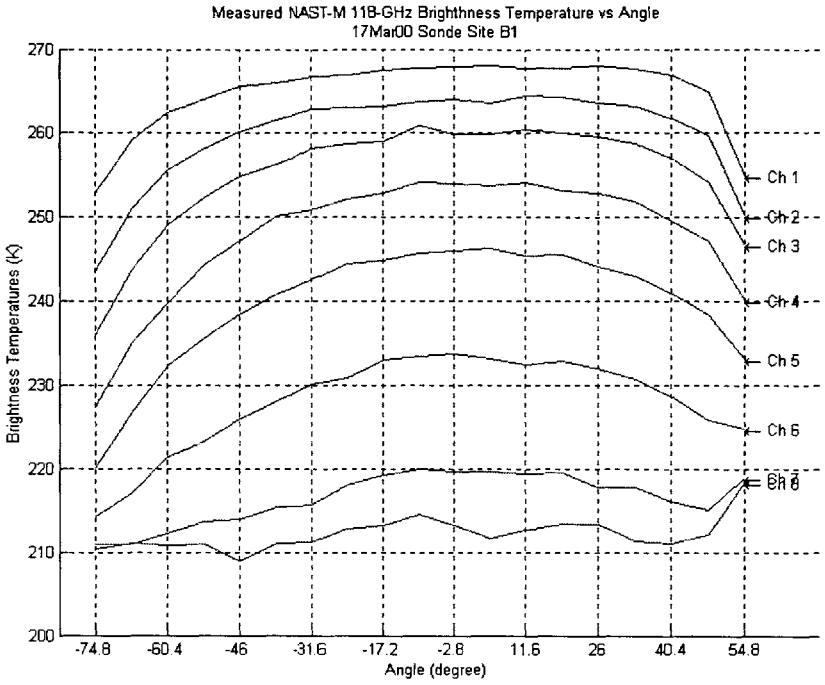


Figure 3-8 Measured 118-GHz data over all spots.

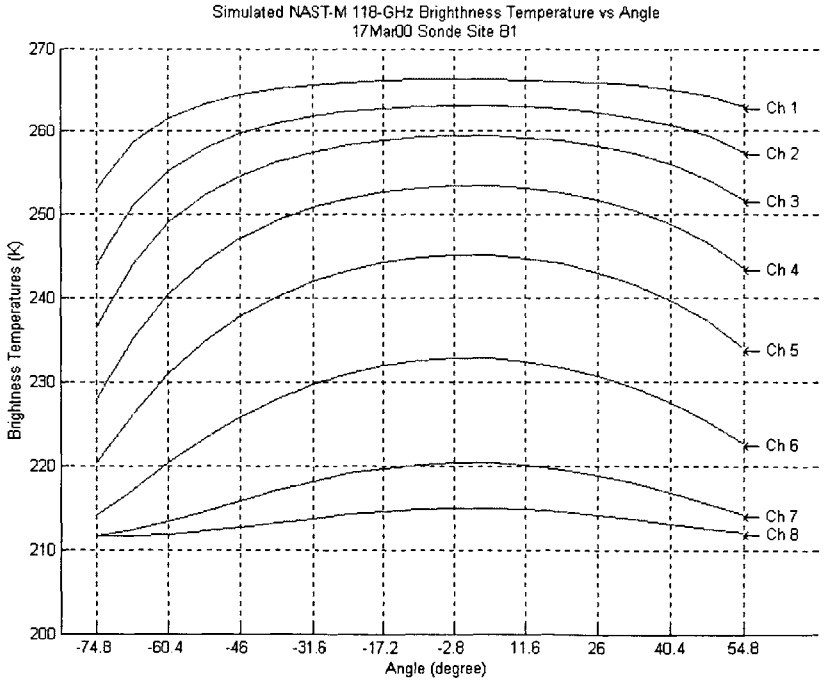


Figure 3-9 Simulated 118-GHz data over all spots.

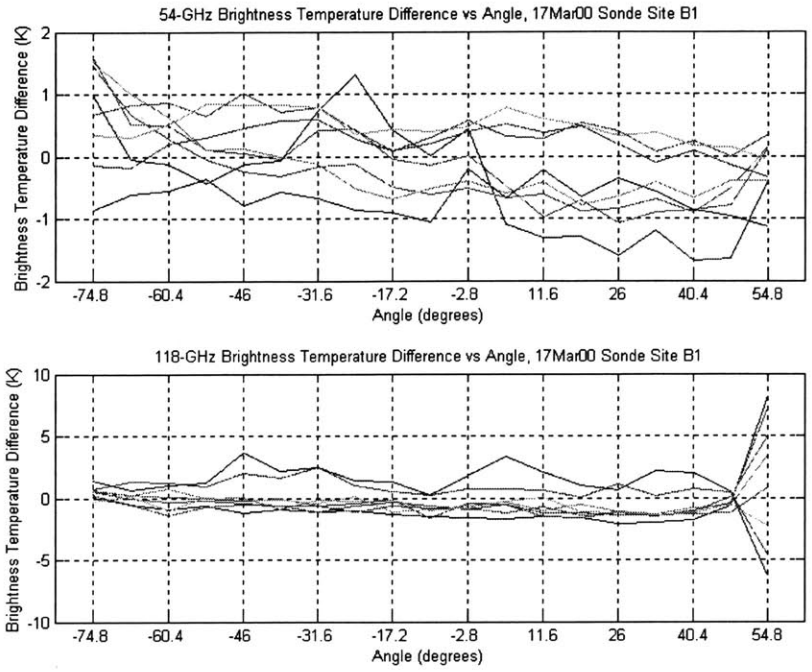


Figure 3-10 Brightness Temperature Difference over all spots.

The flatness of each curve indicates that the offset of measured data from simulated data are similar across each spot. The small slope is the result of a roll from nadir of NAST-M while data was measured. Having shown that the offset is constant over all angles, gain and baseline calculations can be found using the nadir view data.

3.5.1 Calculated Gain and Baseline

Gain and baseline calculations have been found for CLOUDIOP, WVIOP, and AFWEX. During CLOUDIOP, flight dates of Mar 17 00 and Mar 19 00 were chosen gain and baseline calculations. During AFWEX, flight dates of Oct 04 00 and Oct 06 00 were chosen. For the AFWEX deployment, Dec 03 00 and Dec 06 00 were chosen. The results are as follows:

17-Mar-00

Radiosonde Site Location	g54	b54	g118	b118
B1	-0.0056	1.4686	-0.0115	2.0393
B4	-0.0202	4.9562	-0.0260	5.0151
B5	-0.0282	6.9535	-0.0264	5.6557
B6	-0.0450	11.1659	-0.0223	5.1965
C1	-0.0121	3.1224	-0.0380	8.8195
Average:	-0.0222	5.5333	-0.0248	5.3452

Table 3.5 Gain and Baseline Correction for 17 Mar 00.

19-Mar-00

Radiosonde Site Location	g54	b54	g118	b118
B4	-0.1174	28.0051	-0.1130	26.7234
B5	-0.0535	12.8096	-0.0150	3.4008
B6	-0.0442	10.3140	-0.0223	3.3895
C1	-0.0260	6.4525	-0.0037	-0.1242
Average:	-0.0603	14.3953	-0.0385	8.3474

Table 3.6 Gain and Baseline Correction for 19 Mar 00.

4-Oct-00

Radiosonde Site Location	g54	b54	g118	b118
C1	-0.1582	35.3658	-0.1065	23.1515

Table 3.7 Gain and Baseline Correction for 04 Oct 00.

6-Oct-00

Radiosonde Site Location	g54	b54	g118	b118
B4	-0.1402	32.0854	-0.0876	19.5942
B5	-0.0729	16.0016	-0.0526	10.4782
B6	-0.1141	25.6545	-0.072	15.155
C1	-0.1518	34.3656	-0.1445	33.453
Average:	-0.1198	27.0268	-0.0892	19.6701

Table 3.8 Gain and Baseline Correction for 06 Oct 00.

3-Dec-00

Radiosonde Site Location	g54	b54	g118	b118
C1	-0.0522	12.2324	0.0177	-5.1402

Table 3.9 Gain and Baseline Correction for 03 Dec 00.

6-Dec-00

Radiosonde Site Location	g54	b54	g118	b118
C1	-0.0639	15.9054	-0.0247	6.4837

Table 3.10 Gain and Baseline Correction for 06 Dec 00.

Chapter 4

4 Retrieval Algorithm

A custom developed retrieval algorithm for NAST-M is outlined in this chapter. The main goal of the retrieval algorithm is transform nadir viewing brightness temperatures into a temperature profile. An LLSE will be trained for use on data measured during missions CLOUDIOP, WVIOP, and AFWEX.

4.1 Overview of the Algorithm

The retrieval algorithm trained will accept 14 brightness temperatures from NAST-M and one pressure input and produce an estimate of temperatures at 22 atmospheric pressure levels ranging from surface to around 112 mbar. Along with the 22 pressure level temperature retrieval, surface skin temperature, emissivities for the 54-GHz and 118-GHz system and the product of each emissivity with the surface skin temperature will also be attempted for retrieval.

4.1.1 LLSE Retrieval Algorithm

An LLSE will be trained using a MATLAB implemented simulated brightness temperatures of the 54-GHz and 118-GHz system from the TIGR profile set. All eight channels of the 54-GHz system and the first six channels of the 118-GHz system will be used. The last three channels of the 118-GHz are unusable because of interference from the LO. The pressure level at which the brightness temperatures are simulated will be used to position NAST-M above the Earth's surface. The simulated brightness temperatures and pressure are inputs into the LLSE. As for outputs, the first 22 pressure levels of the TIGR profile set will be retrieved with the associated surface temperature, surface emissivity, and the product of surface temperature and surface emissivity. All 1761 profiles in the TIGR set will be used in finding the best LLSE. Overall, the LLSE will accept 15 inputs and estimate 27 outputs.

4.1.2 Finding an LLSE

Using the models of surface emissivity, surface temperatures, nominal pressure of NAST-M's location, and the pre-computed brightness temperature components, 47,547 brightness temperatures were simulated. The surface parameters are be broken up into categories. Table boo shows how they were broken up.

Surface Emissivity			Surface Temperature			Observation Pressure		
Min	In Between	Max	Min	In Between	Max	Min	In Between	Max
0.95	unif(0.95,1.0)	1	Ts - 3	Ts + N(0,3)	Ts + 3	86.07	unif(86.07,106.27)	106.27

Table 4.1 Range of Surface Parameters.

For each profile, the minimum surface emissivity value was used along with the minimum surface temperature and minimum pressure. Brightness temperature is then simulated. Then a random surface emissivity in range of 0.95 to 1.0 was generated while keeping the other parameters the same. Finally the maximum value of surface emissivity is used while keeping the other values the same. The same method is applied for the surface temperature and the observation pressure. The result is a super set of simulated brightness temperatures from the TIGR set. The tree diagram (Figure 4-1) is another interpretation of how the super set is formed.

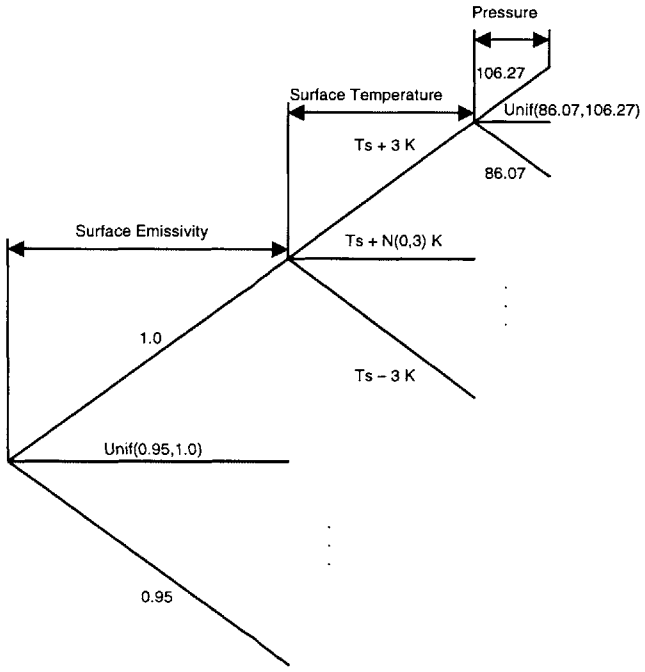


Figure 4-1 Super Set of Simulated TIGR Brightness Temperatures.

Following the tree from surface emissivity to surface temperature to pressure represents 1761 simulated brightness temperatures. There are 27 different paths that lead to a new realization of 1761 simulated brightness temperatures. Only the first 14 channels of simulated brightness temperatures are to be used. Now the pressure is concatenated onto the brightness temperatures and this becomes the input X. Using the instrument sensitivity table (Table 3.3), instrument noise, N, can be simulated using each channels standard deviation. The noise is assumed to be zero-mean Gaussian. The noise realization is then added. The sample mean or $E[X]$ is now calculated for each row of the input. The sample covariance matrix, K_{XX} of X is calculated. The noise covariance matrix, K_{NN}, is simply a diagonal matrix with the variance of each channel on its diagonal.

For each input, simulated from a TIGR profile, the corresponding 22 pressure levels of that profile is the output. The generated surface parameters are concatenated to the temperature profiles and becomes the output Y. Again the sample mean or $E[Y]$ is calculated for each row of the output. The sample cross covariance matrix, K_{YX}, between the output and input is then calculated. The LLSE is:

$$\hat{Y}(X) = E[Y] + K_{YX} \cdot (K_{XX} + K_{NN})^{-1} \cdot (X + N - E[X]) \qquad (4.1)$$

For any subsequent input with M observations, the output will have M number of retrieved temperature profiles.

4.1.3 Results

To test performance, randomly generate a new realization of simulated brightness temperatures and noise from the TIGR set and the small ARM ensemble collected. This time only one path from Figure 4-1 should be used. The randomly generated parameters are the best case to use. Estimate a temperature profile using the LLSE. Perform RMSE at all pressure levels then plot. With the estimated temperature profiles along with the surface parameters that were estimated, use the same pressure as before and re-simulate brightness temperature. Perform RMSE with the

brightness temperatures. The closed loop analysis will give an indication of how well the LLSE will perform.

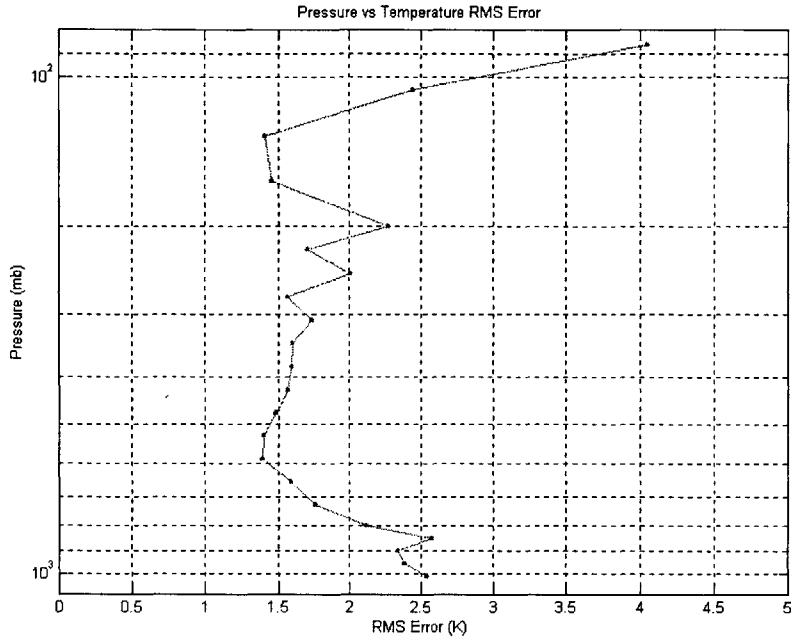


Figure 4-2 RMS Retrieval Error of the TIGR Ensemble.

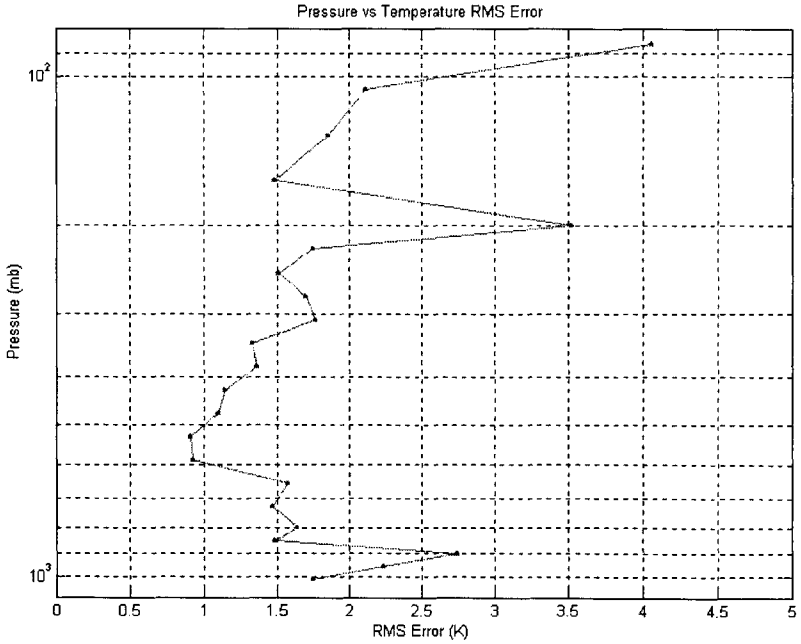


Figure 4-3 RMS Retrieval Error of the ARM Ensemble.

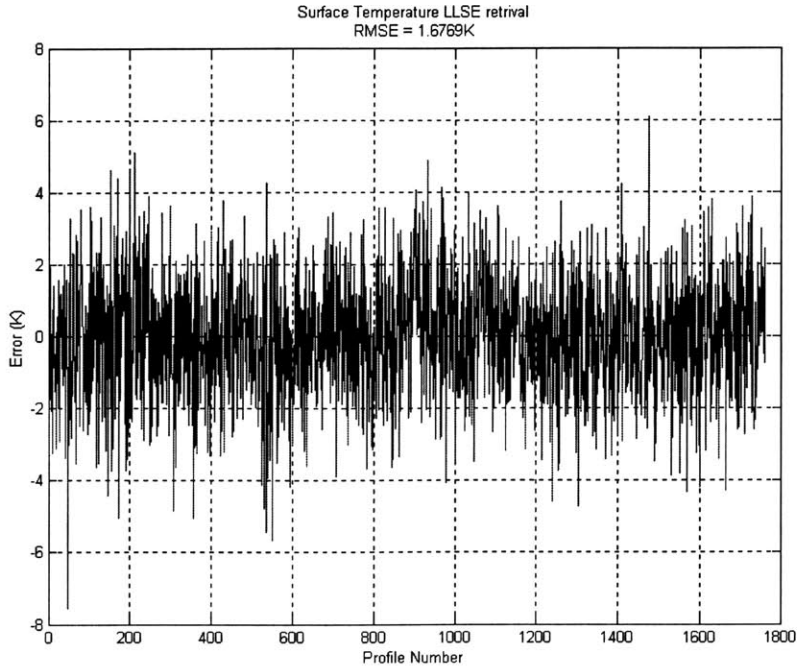


Figure 4-4 Surface Temperature Retrieval Error on TIGR.

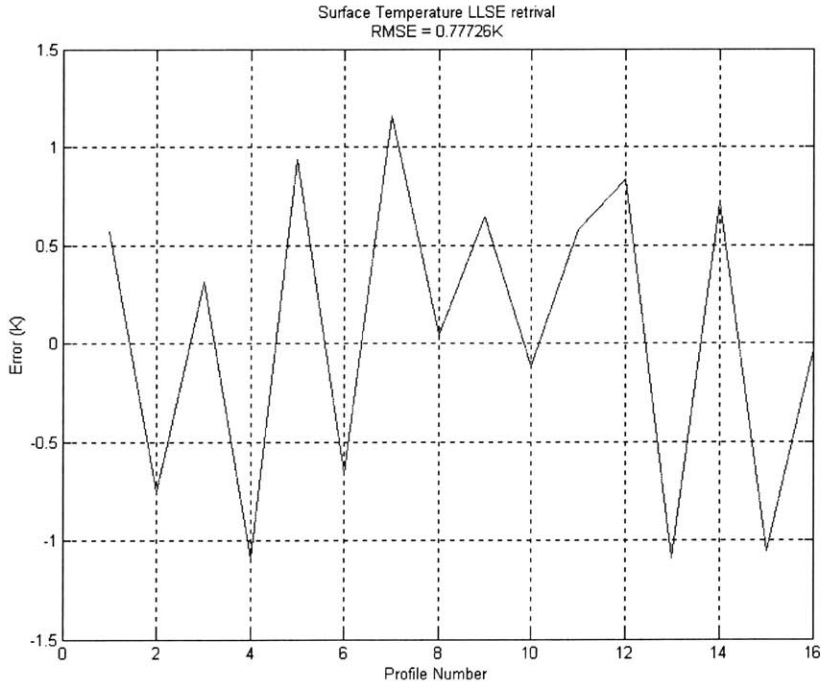


Figure 4-5 Surface Temperature Retrieval Error on ARM.

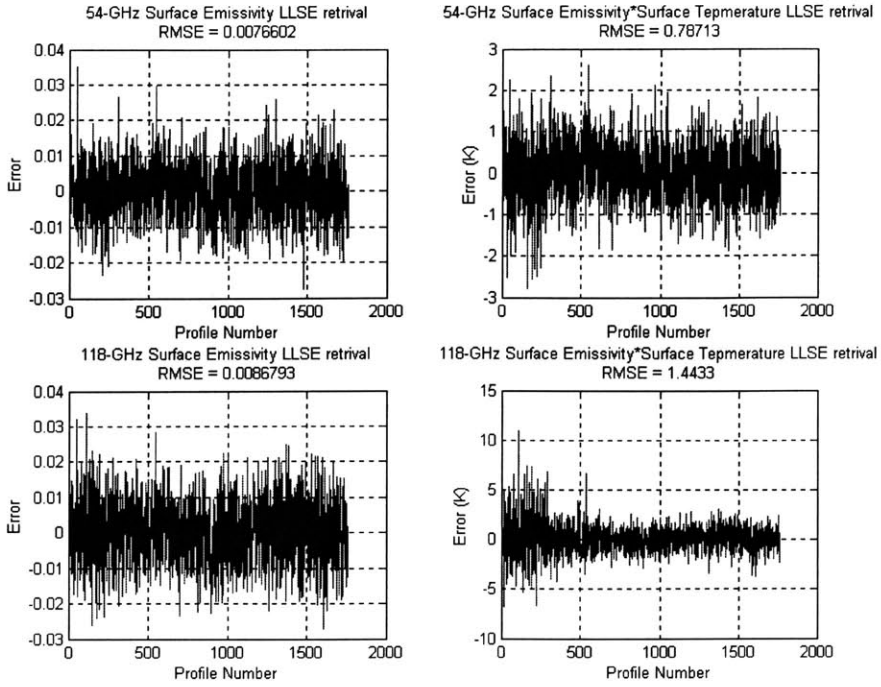


Figure 4-6 Surface Emissivity and Product of Surface Emissivity with Surface Temperature Retrieval Error on TIGR.

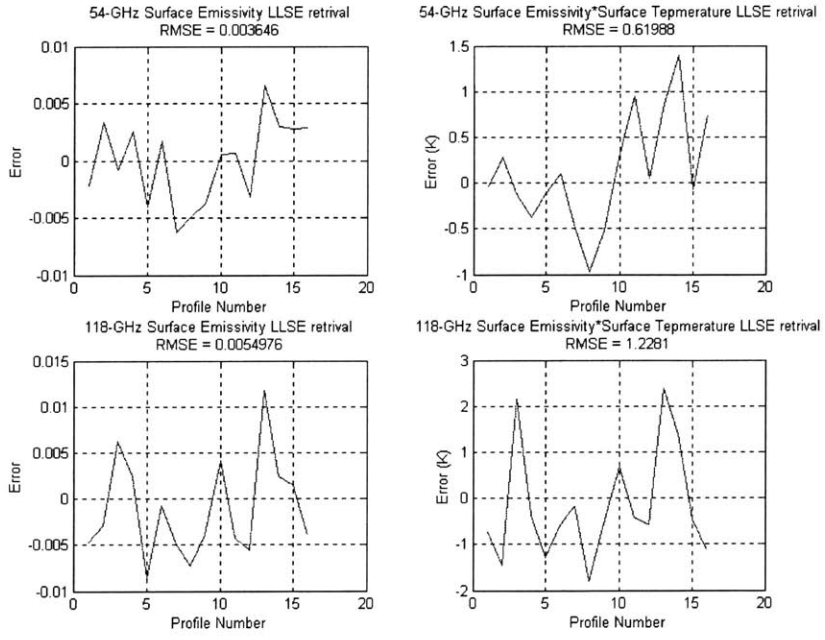


Figure 4-7 Surface Emissivity and Product of Surface Emissivity with Surface Temperature Retrieval Error on ARM.

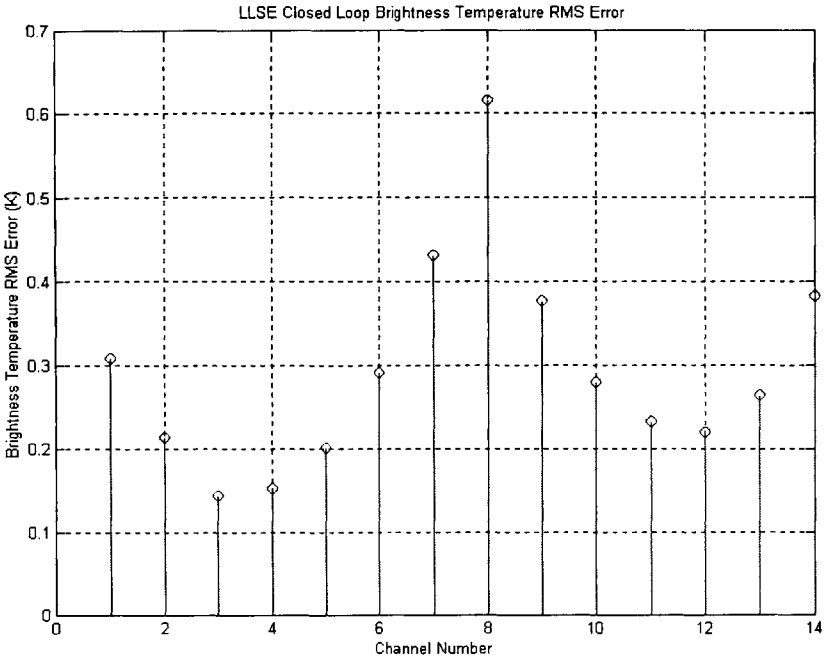


Figure 4-8 RMS Error of Closed Loop Test on TIGR.

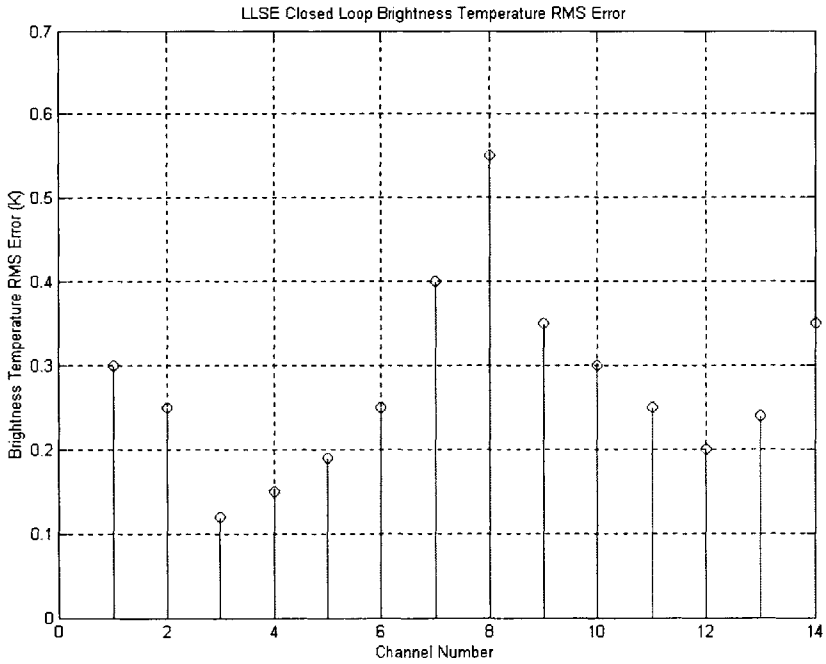


Figure 4-9 RMS Error of Closed Loop Test on ARM.

Chapter 5

5 NAST-M CLOUDIOP, WVIOP, and AFWEX Retrievals

The results of the temperature profile retrievals from flight deployments CLOUDIOP, WVIOP, and AFWEX are listed in this chapter. During the CLOUDIOP deployment, retrievals were performed using NAST-M measured data for March 17, 00 and March 19, 00. Data collected on October 04, 00 and October 06, 00 from the WVIOP deployment has also been attempted for temperature retrievals. Finally, data from AFWEX on 03 December, 00 and 06 December, 00 has been attempted for retrieval. All three missions took place over the Southern Great Plains area over Kansas and Oklahoma. To test the validity of NAST-M temperature retrievals, radiosondes collected on the particular days from the ARM program were used for comparison. Due to aircraft roll or other reasons, artifacts may be presents. Further study may be required for a more accurate retrieval.

5.1 CLOUDIOP Retrievals

From start to finish, the CLOUDIOP mission started on March 11, 00 and ended on March 21, 00. The days chosen for temperature retrievals were March 17, 00 and March 19, 00 because of the number of radiosondes collected for those days. Table 5.1 is a compiled flight log consisting of details from the mission such as weather, instrument performance, and number of ARM radiosondes collected.

Date	Start (GMT)	End (GMT)	Sondes	Weather	Comments
3/11	17:20	20:10	0		Engineering flight 1
3/13	19:00	23:15	0	Mostly clear	Engineering flight 2
3/14	18:00	21:30	1		Ferry flight
3/17	18:30	23:00	10	Partly cloudy	
3/19	15:30	20:30	8	Partly cloudy	
3/20	14:00	17:00	2	Scattered Clouds	
3/21	12:30	16:30	2		Ferry flight

Table 5.1 CLOUDIOP 2000 NAST-M Flight Summary.

5.1.1 CLOUDIOP Retrieval Results

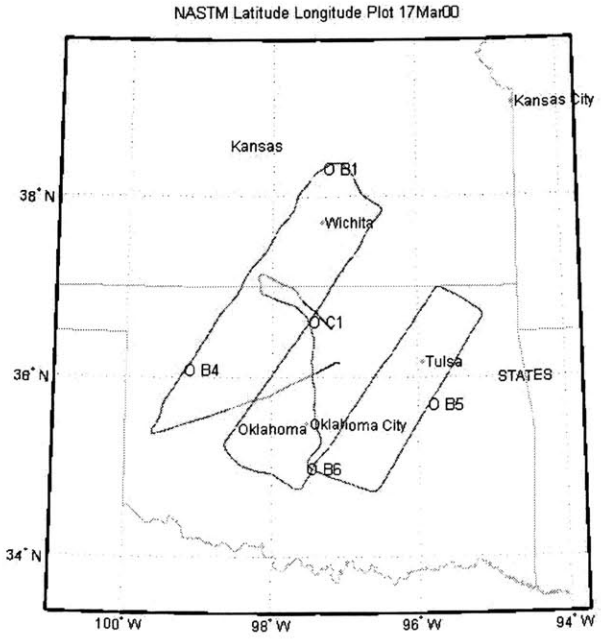


Figure 5-1 17Mar00 Flight Path.

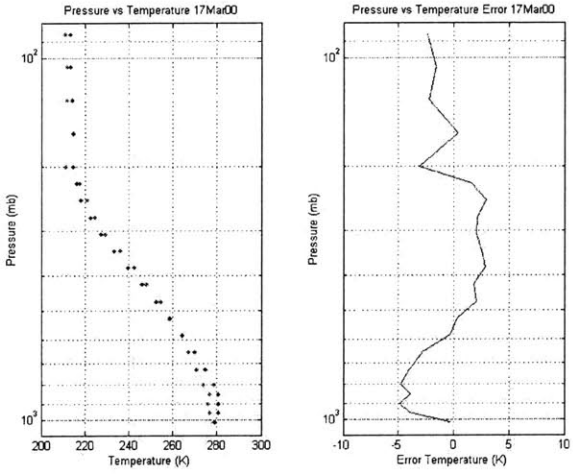


Figure 5-2 17Mar00 Temperature Retrieval Comparison for index 1300.

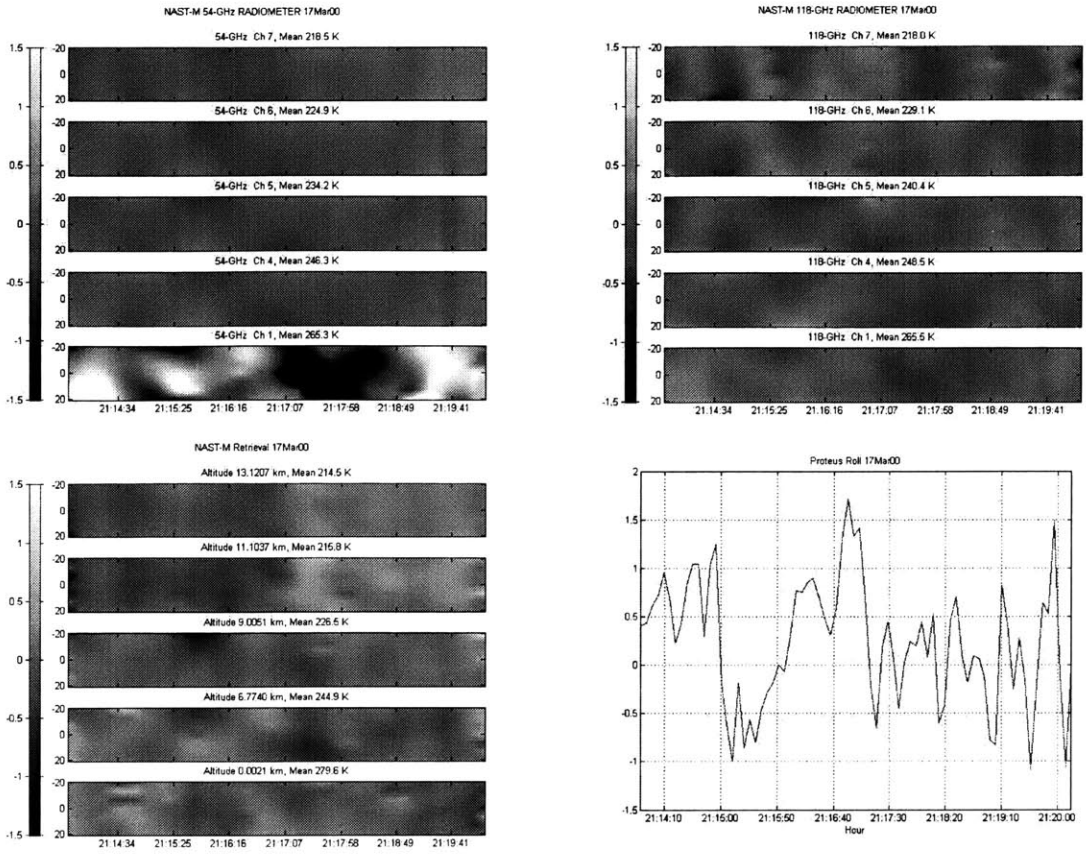


Figure 5-3 17Mar00 Temperature Imagery Comparison for index 1300.

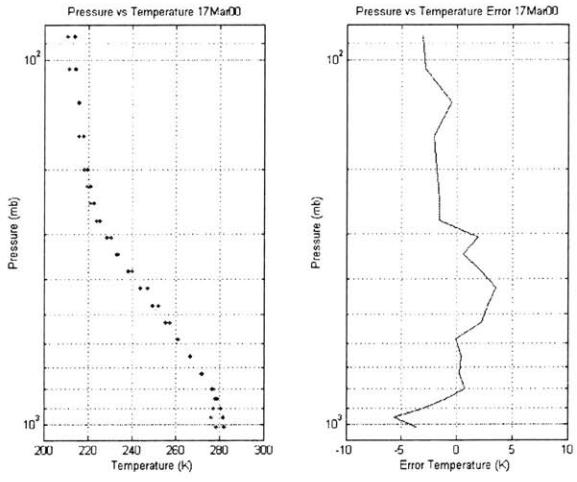


Figure 5-4 17Mar00 Temperature Retrieval Comparison for Index 2370.

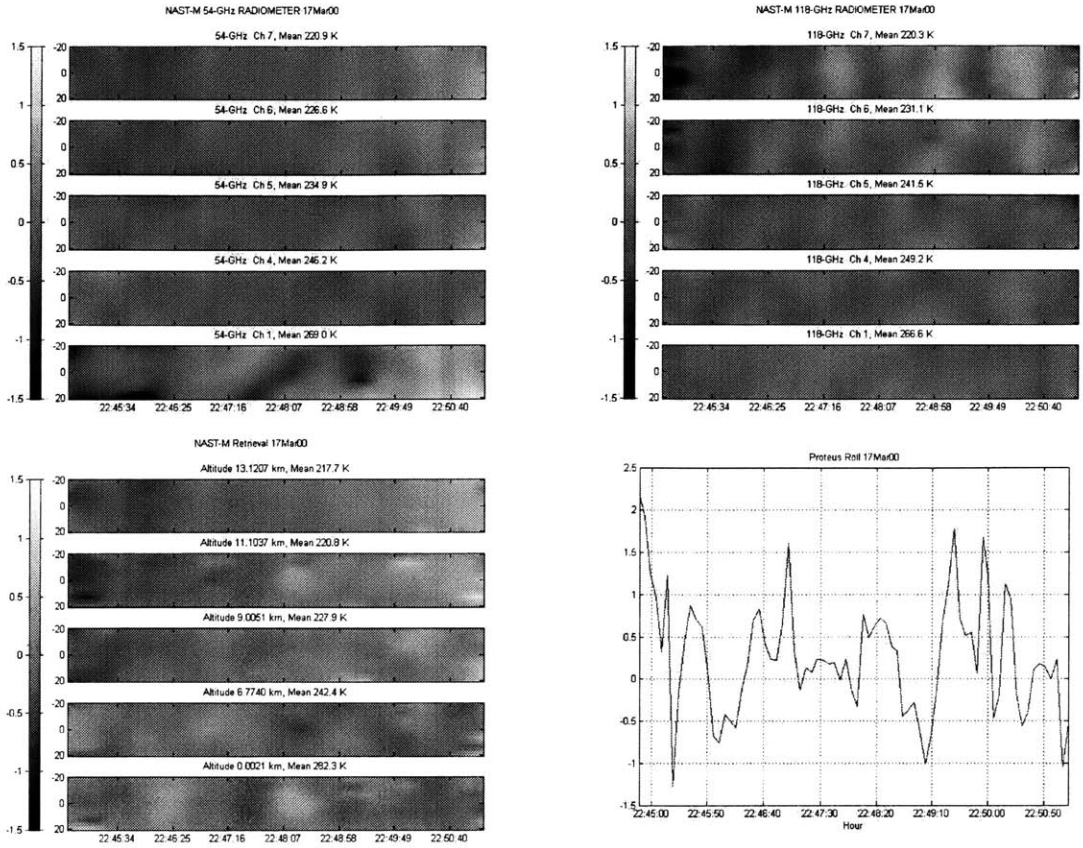


Figure 5-5 17Mar00 Temperature Imagery for index 2370.

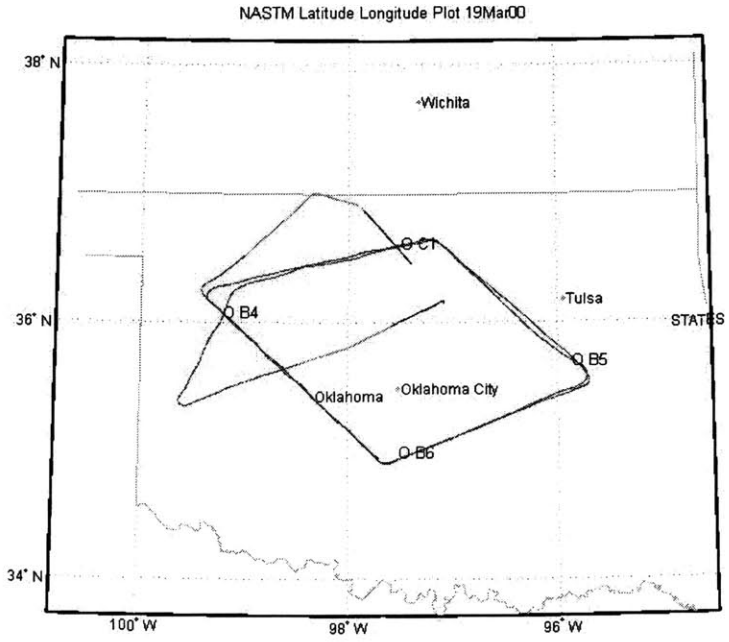


Figure 5-6 19Mar00 Flight Path.

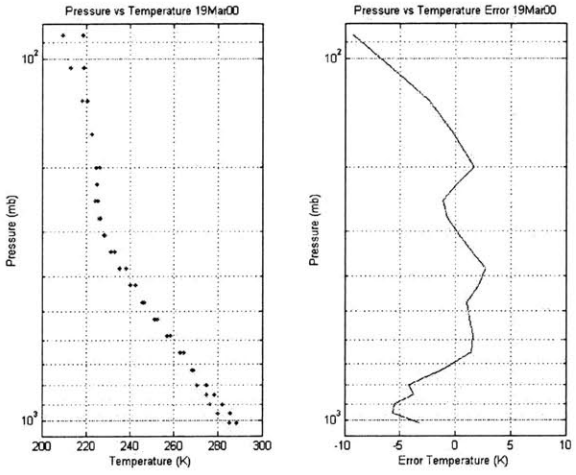


Figure 5-7 19Mar00 Temperature Retrieval Comparison for index 1620.

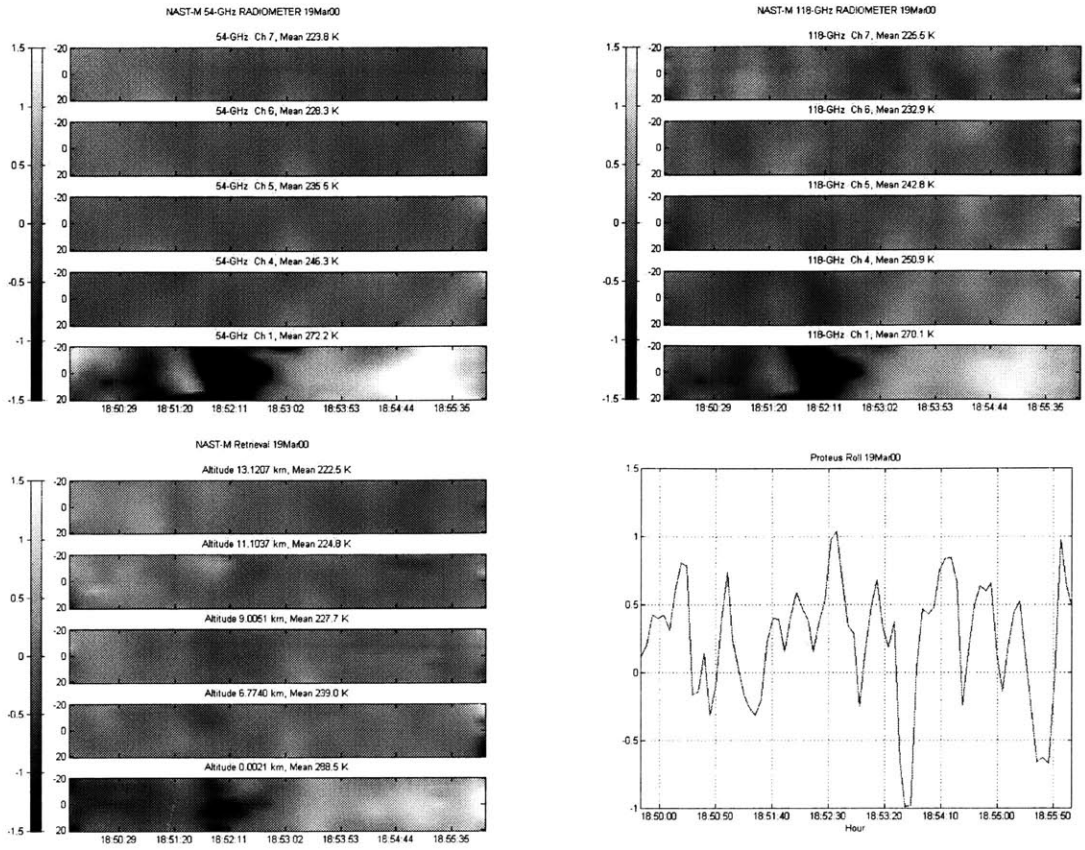


Figure 5-8 19Mar00 Temperature Imagery for index 1620.

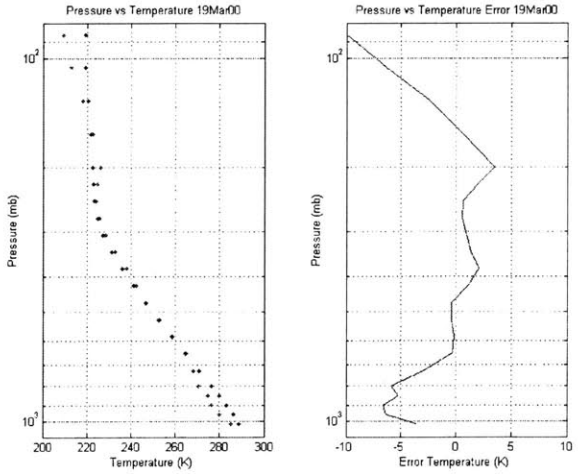


Figure 5-9 19Mar00 Temperature Comparison for index 2915.

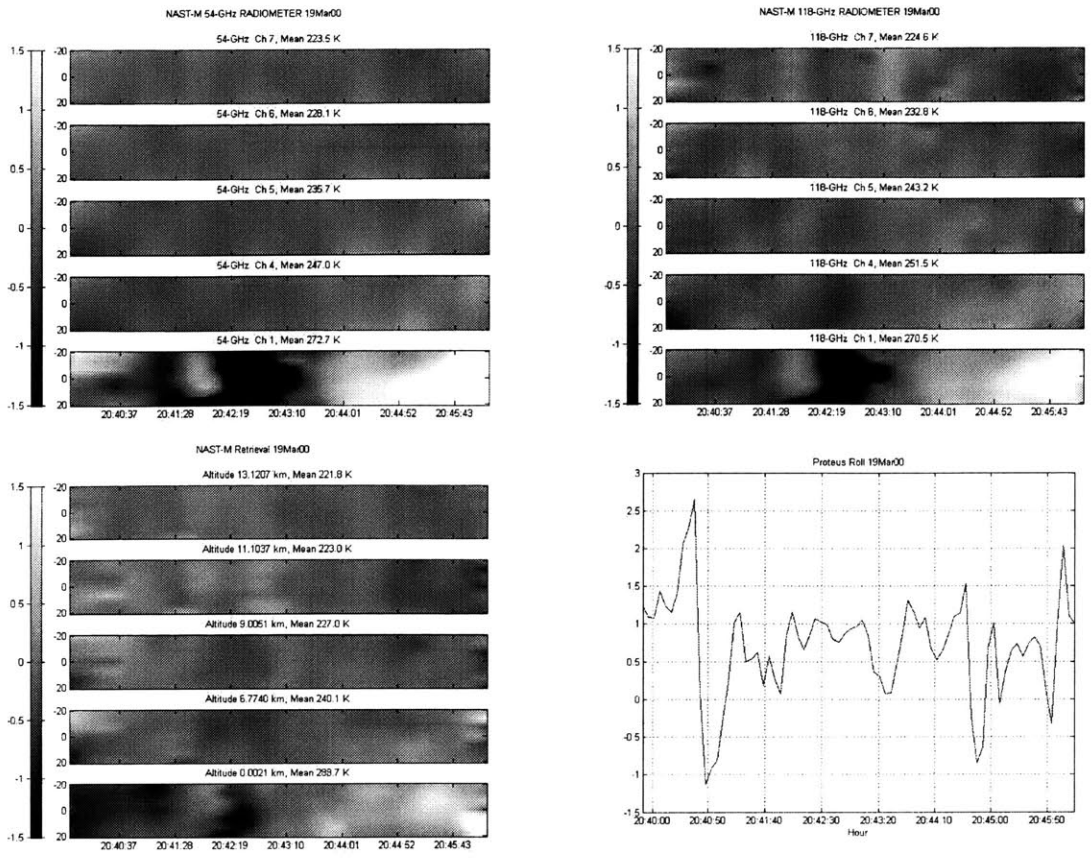


Figure 5-10 19Mar00 Temperature Imagery for index 2915.

5.2 WVIOP Retrievals

WVIOP mission lasted from September 30, 00 to October 6, 00. Two days chosen for retrieval were October 4, 00 and October 6, 00. This particular mission fell short due to instrument failure.

Date	Start(GMT)	End(GMT)	Sondes	Comment	Weather
9/30	14:00	18:00	0	Engineering Flight	
10/1	15:30	19:00	1	Ferry flight	
10/4	14:00	20:00	3		
10/6	14:30	17:00	4	Instrument failure	Partly cloudy

Table 5.2WVIOP 2000 NAST-M Flight Summary.

5.2.1 WVIOP Retrieval Results

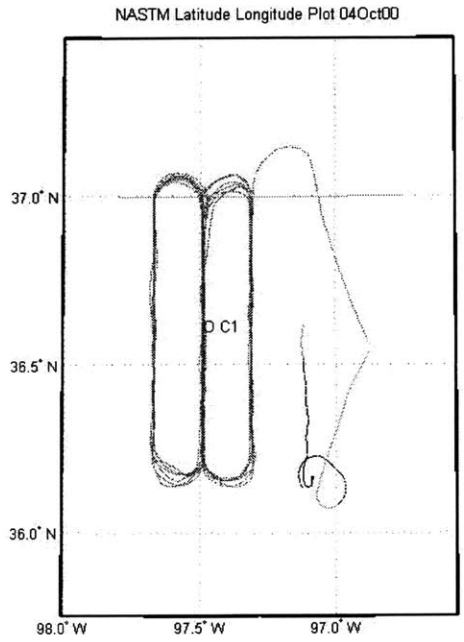


Figure 5-11 04Oct00 Flight Path.

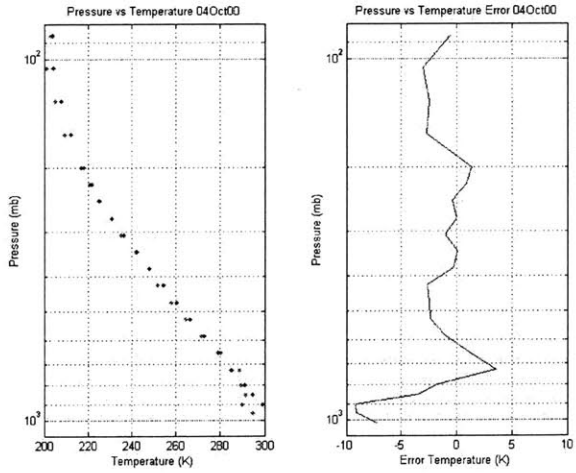


Figure 5-12 04Oct00 Temperature Retrieval Comparison for index 1600.

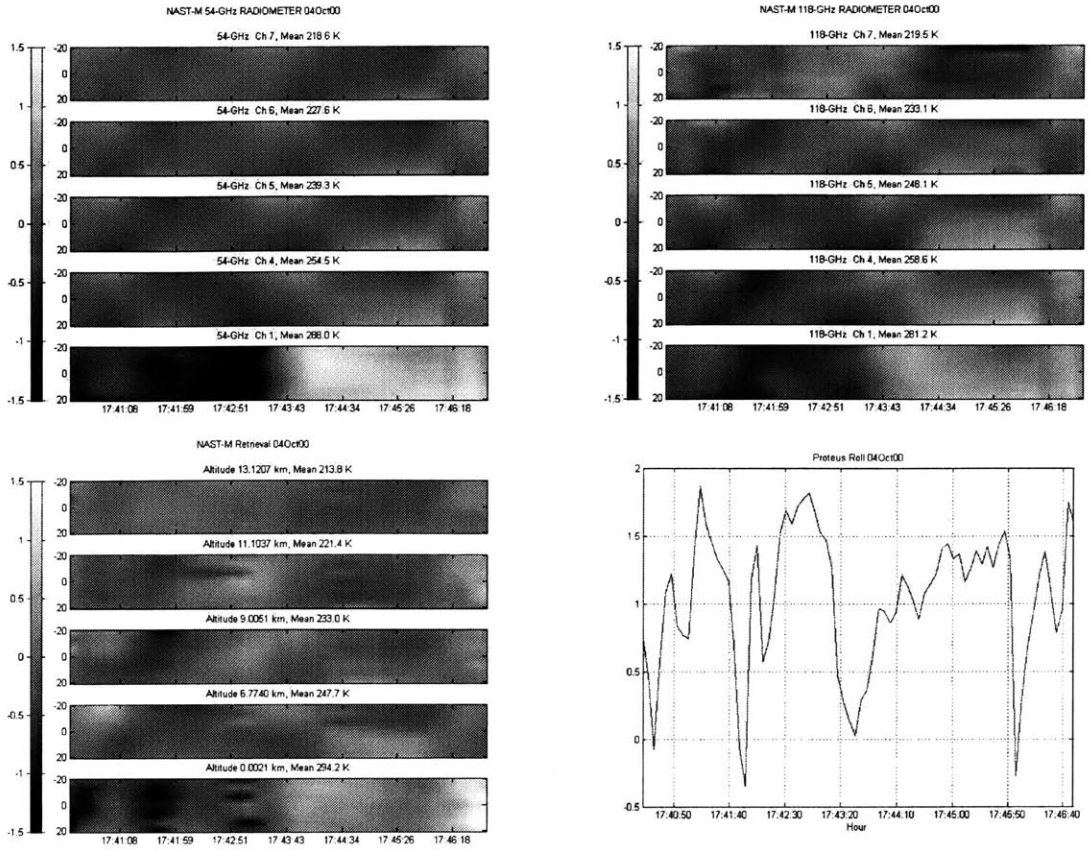


Figure 5-13 04Oct00 Temperature Imagery for index 1600.

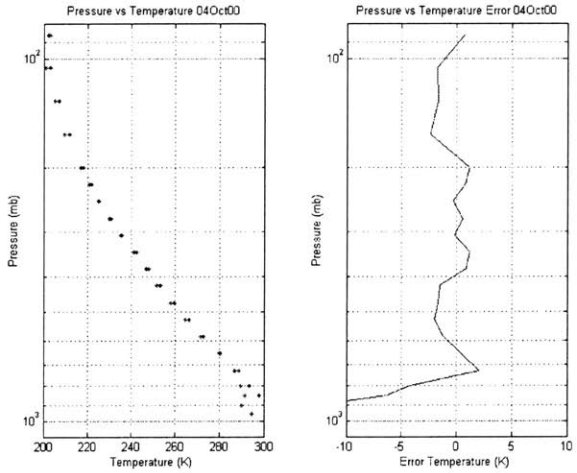


Figure 5-14 04Oct00 Temperature Retrieval Comparison for index 2795.

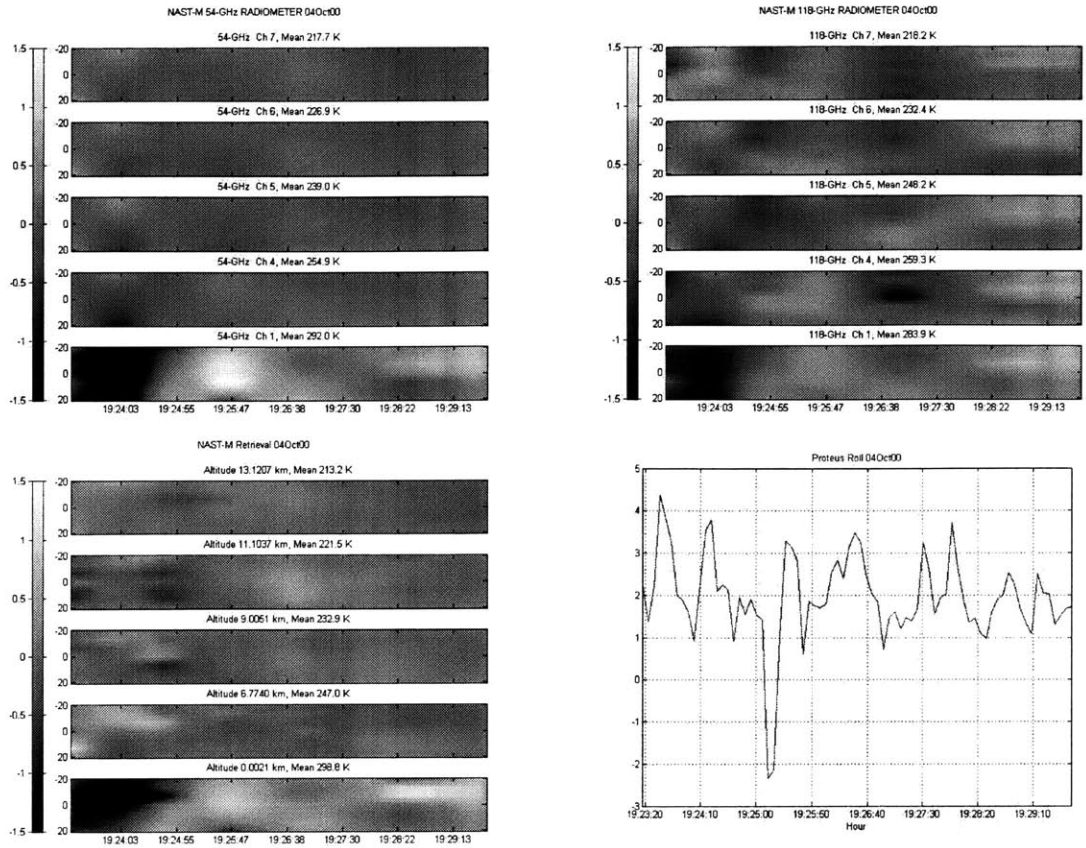


Figure 5-15 04Oct00 Temperature Imagery for index 2795.

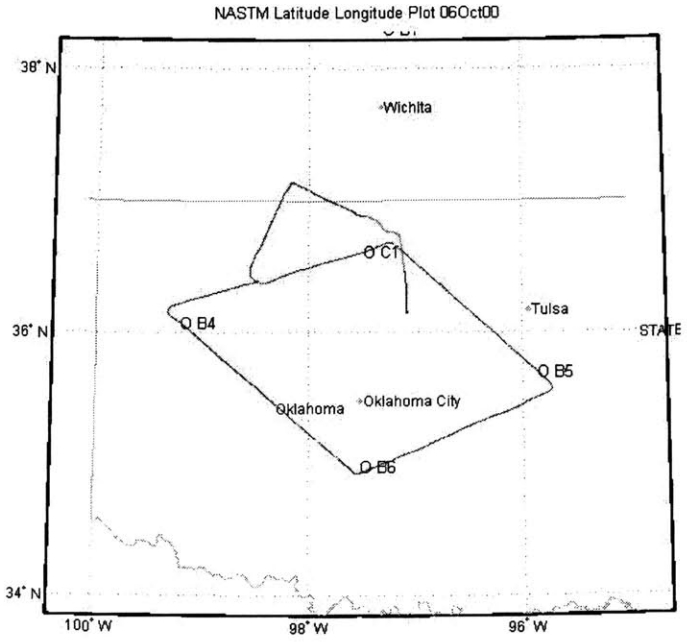


Figure 5-16 06Oct00 Flight Path.

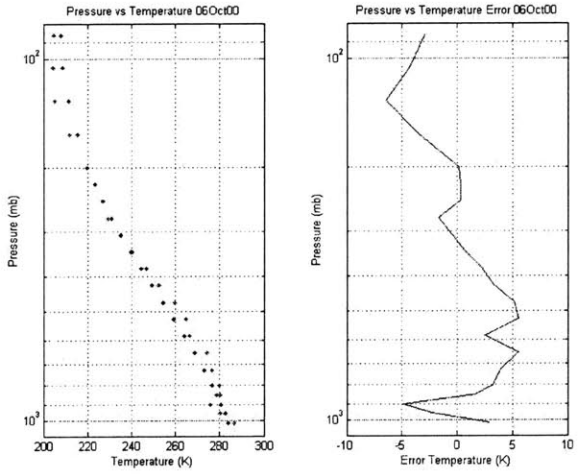


Figure 5-17 06Oct00 Temperature Retrieval Comparison for index 1110.

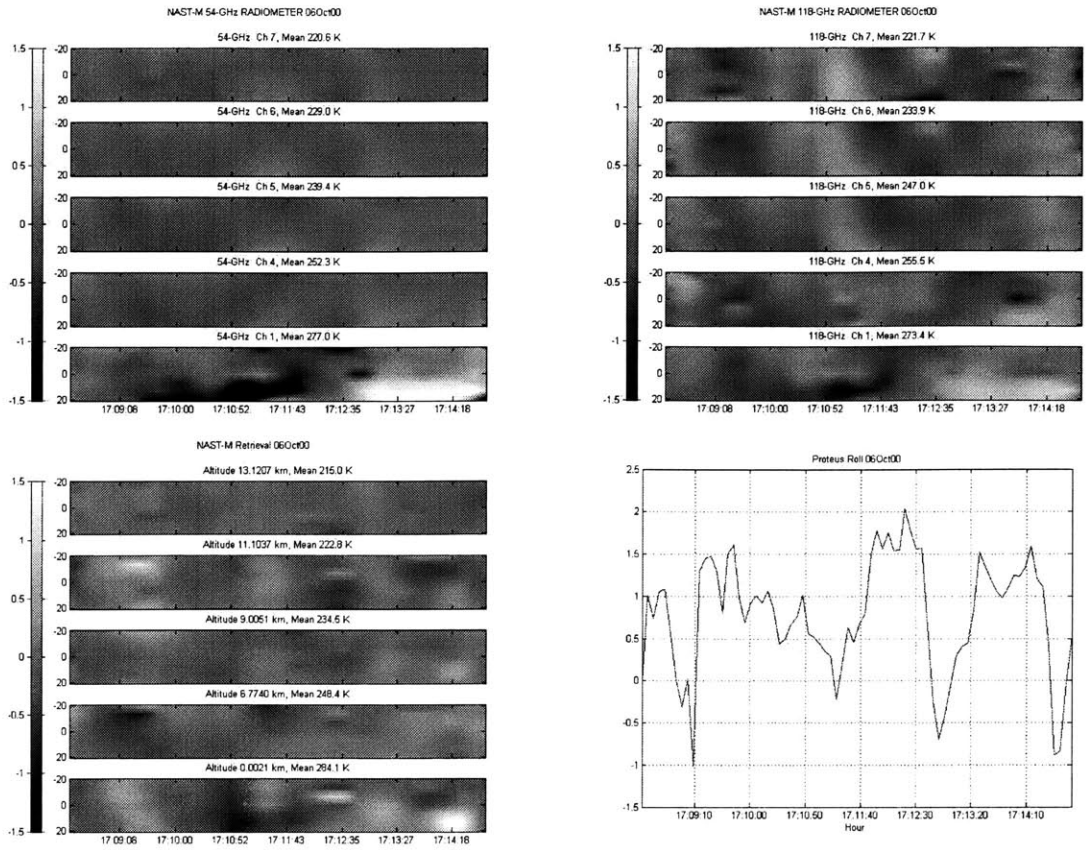


Figure 5-18 06Oct00 Temperature Imagery for index 1110.

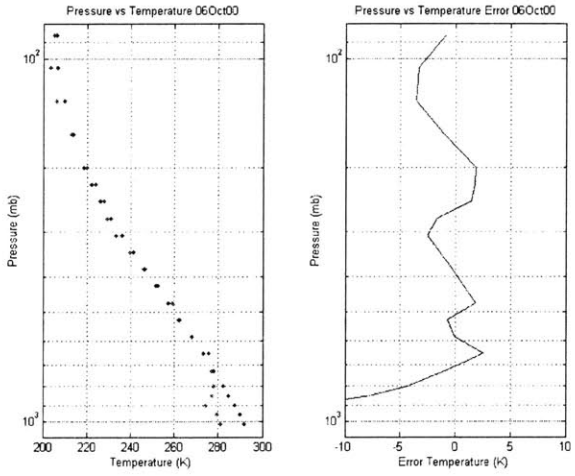


Figure 5-19 06Oct00 Temperature Retrieval Comparison for index 1608.

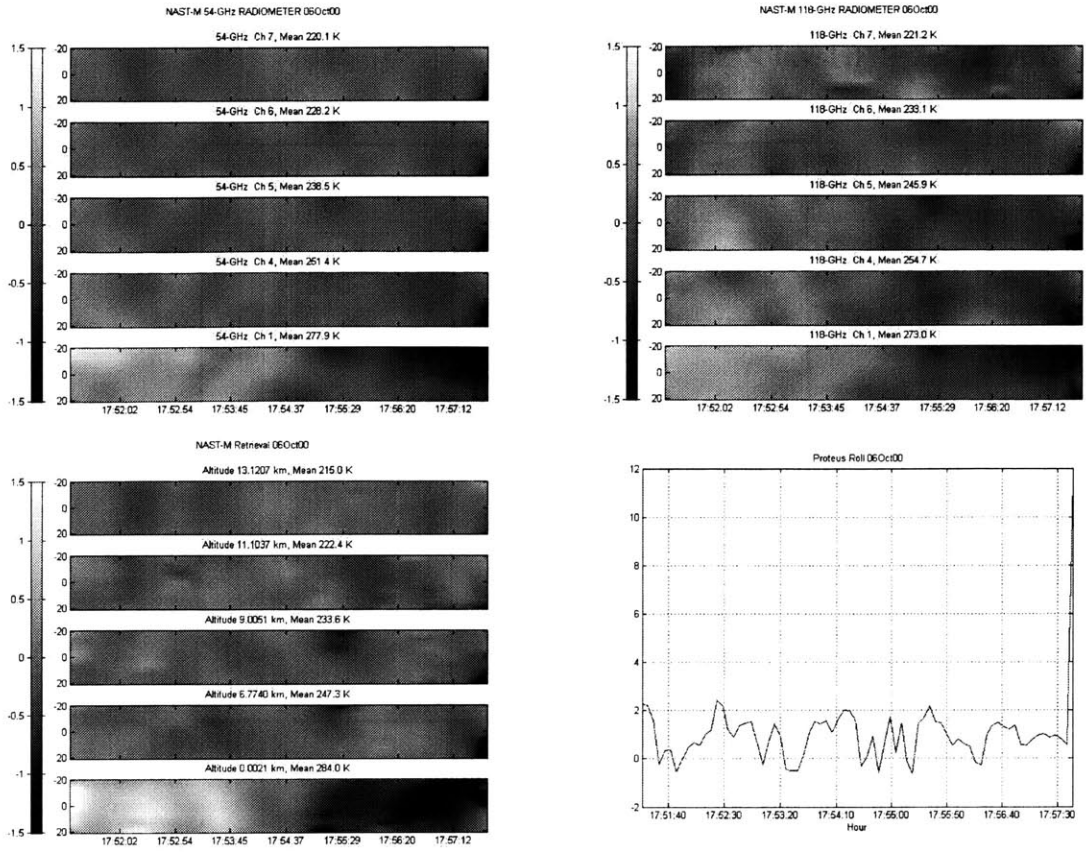


Figure 5-20 06Oct00 Temperature Imagery for index 1608.

5.3 AFWEX Retrievals

The AFWEX mission deployed November 27, 00 and ended December 08, 00. Proteus data was not available for this deployment. Snapshot images were also not available. For the entire duration of this mission, Channel 3 of the 54-GHz system was malfunctioning. As a result an LLSE was implemented without data from Channel 3. Temperature retrievals were attempted on December 3, 00 and December 6, 00.

Date	Start(GMT)	End(GMT)	Sondes	Comments	Weather
11/27	20:30	23:00	0	Ferry Flight, Inst Failure	
12/1	22:00	2:15	4	Channel 3 malfunction	
12/3	0:30	4:30	1	12-4-00 according to GMT	
12/6	2:15	6:30	2	12-7-00 according to GMT	Mostly Cloudy
12/7	22:30	2:15	2		
12/8	0:15	7:30	6	12-9-00 Ferry flight Bad Data	

Table 5.3 AFWEX 2000 NAST-M Flight Summary.

5.3.1 AFWEX Retrieval Results

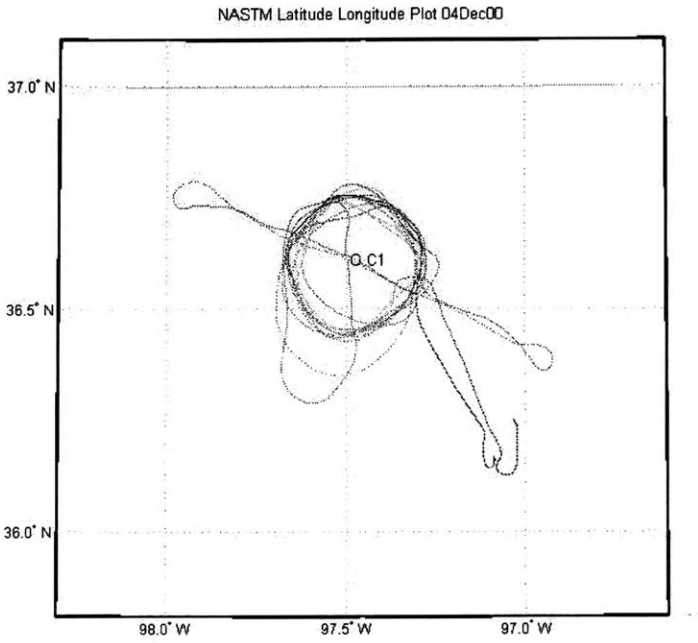


Figure 5-21 03Dec00 Flight Path.

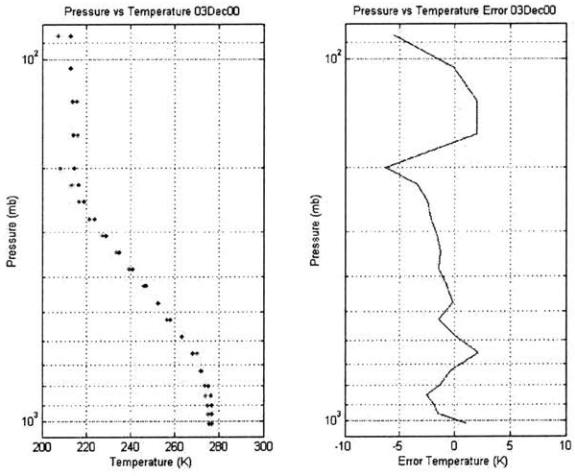
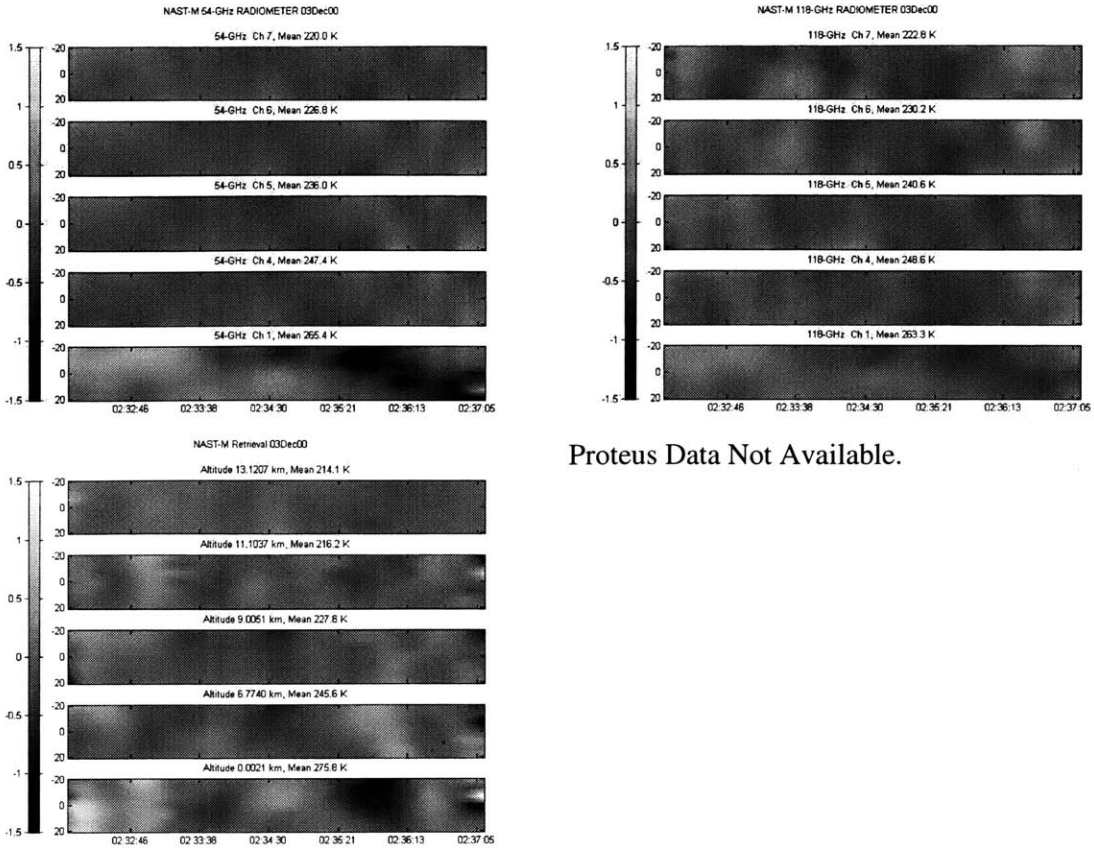


Figure 5-22 03Dec00 Temperature Retrieval Comparison for index 841.



Proteus Data Not Available.

Figure 5-23 03Dec00Temperature Imagery for index 841.

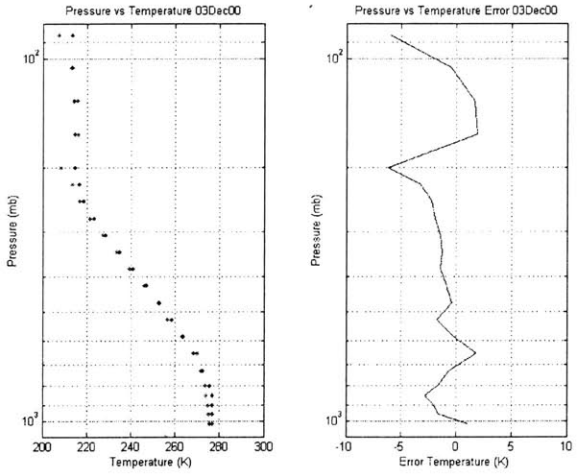
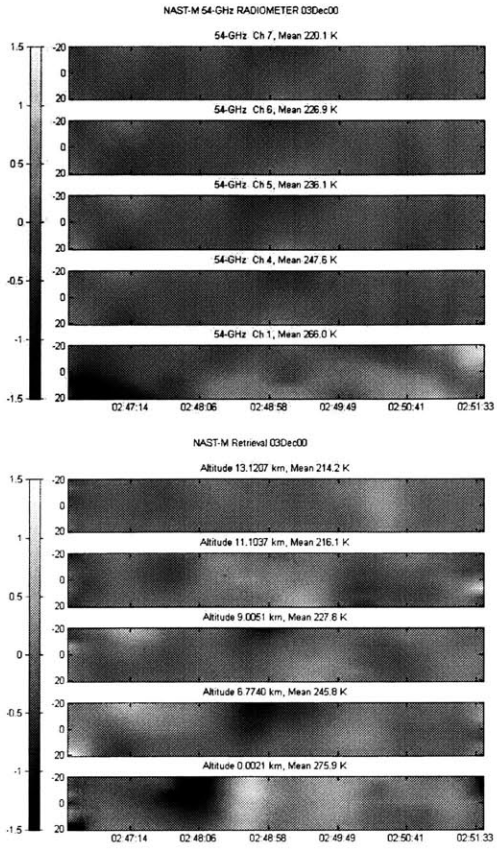
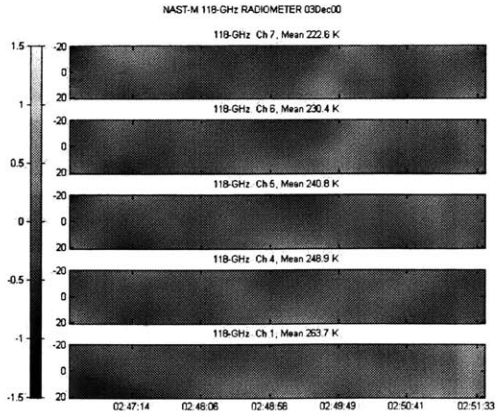


Figure 5-24 03Dec00Temperature Retrieval Comparison for index 1009.





Proteus Data Not Available.

Figure 5-25 03Dec00Temperature Imagery for index 1009.

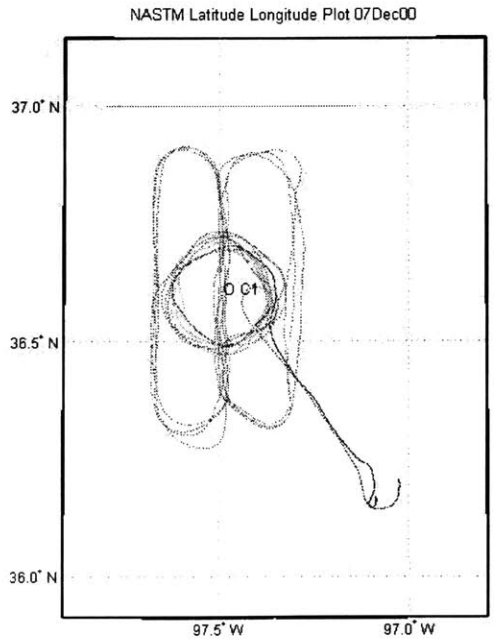


Figure 5-26 06Dec00 Flight Path.

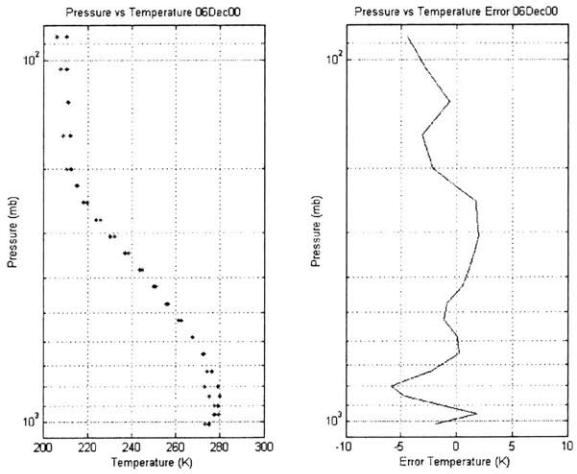
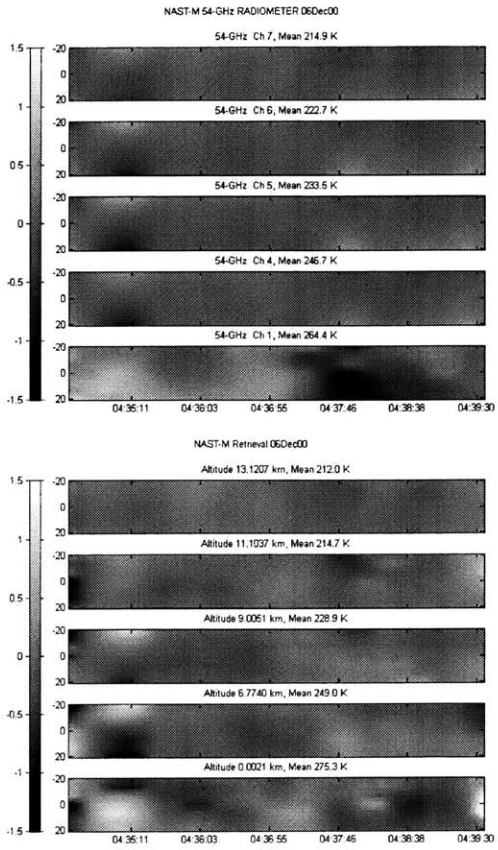
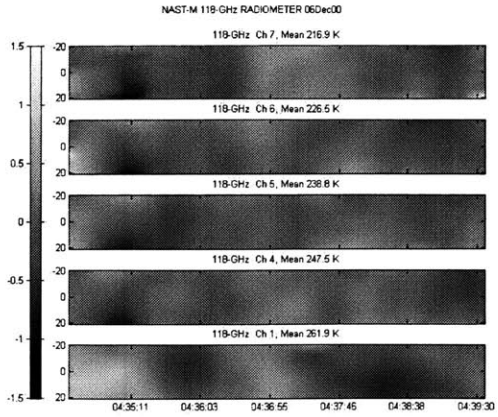


Figure 5-27 06Dec00 Temperature Retrieval Comparison for index 1561.





No Proteus Data Available.

Figure 5-28 06Dec00 Temperature Imagery for index 1561.

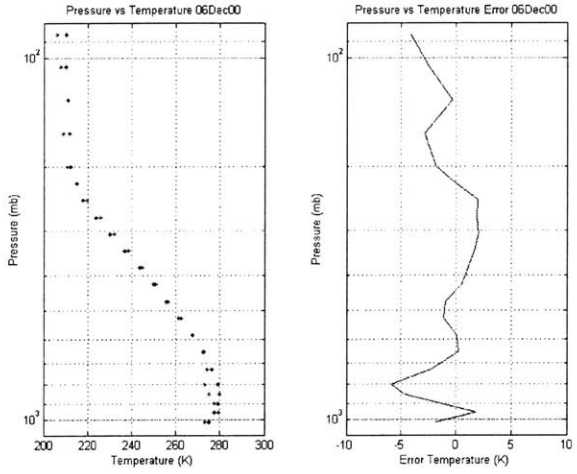
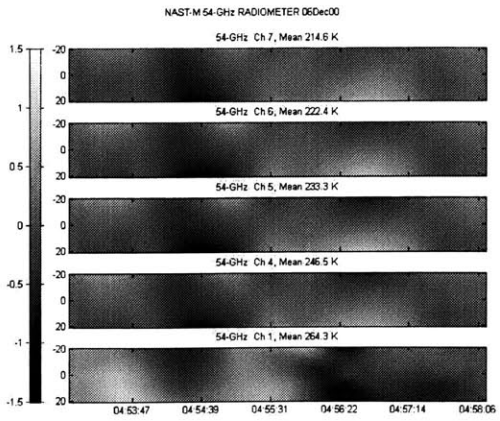
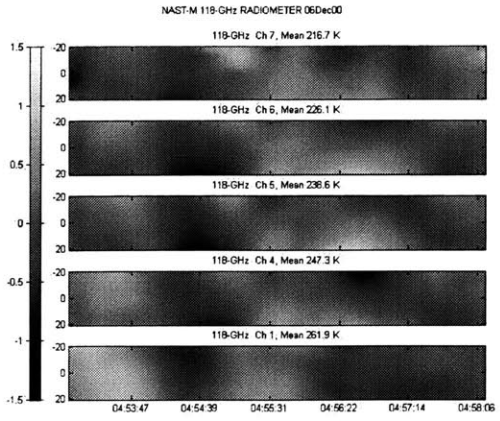
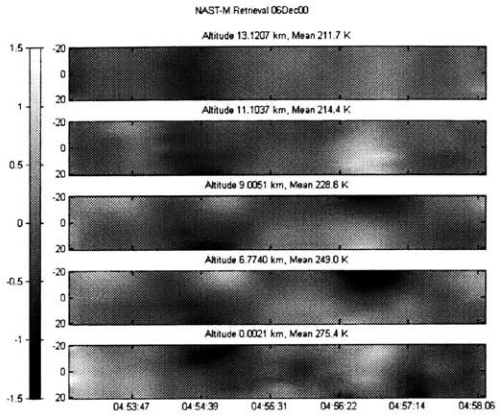


Figure 5-29 06Dec00 Temperature Retrieval Comparison for index 1777.







No Proteus Data Available.

Figure 5-30 06Dec00 Temperature Imagery for index 1777.

Chapter 6

6 Conclusions and Future Work

In attempting to retrieve temperature profiles, the LLSE has been shown to perform well. Using the entire TIGR ensemble for brightness temperature simulations, the average RMS error throughout each pressure level in retrieving temperature profiles is ~2K. During the simulation of brightness temperatures from the ensemble, instrument noise is simulated and added. The addition makes up part of the 2K rms error. A very small portion is contributed by the use of partial atmosphere instead of full atmosphere for brightness temperature simulation. Performing a closed looped test revealed an average rms error of ~0.3K from the original simulated brightness temperatures and re-simulated brightness temperatures from retrieved temperature profiles.

The use of gain and baseline correction for calibration has been successful. The most success occurs for clear air. Mostly cloudy days result in a skew for temperature retrievals when comparing to radiosonde data. The issue of using a single set of gain and baseline calculations, as opposed to a multiple set for each spot, has been reconciled by plotting the difference between gain and baseline modified NAST-M data and simulated brightness temperatures from a nearby radiosonde for particular days. The flatness of each channel plot across all spots has been reassuring that one gain and baseline calculation with nadir data can be used on spots near nadir.

With the use of brightness temperature imagery, signs of external interference or noise in the 118-GHz system have been made apparent. Spatially low pass filtering the image helped in removing noise. Possible natural phenomenons in the atmosphere are left behind. When the Proteus has the opportunity to fly over the same atmosphere, perturbations that appear to be natural can be verified. Mar19, 2000 is an example of verifying a natural phenomenon by the use of a second over pass. Using the Proteus roll data, artifacts can be ruled out as well. Plotting RMS differences between channels 6 and 7 of the 54-GHz and 118-GHz, it has been shown that the 118-GHz system has a tendency to "noisy". Because of the noisiness of the 118-GHz system, only 54-GHz data from NAST-M have been used for temperature retrievals. Excluding channel

1, Perturbations from the mean in the 54-GHz system and 118-GHz system can be squeezed into a ± 0.5K range and ± 0.8K range respectively.

From the temperature retrieval imagery, it has been shown that the average rms perturbation from the mean of each pressure level is ~0.5k. Since only 54-GHz data have been used for retrievals, the images retrieved are similar. However, there were cases of the temperature retrieved images that were similar to the 118-GHz brightness temperature images.

6.1 Future Work

The calibration algorithm currently in use can be upgraded to help minimize the effects of outside interference on the 118-GHz system. Interference shows up as vertical stripes in the brightness temperature images. An additional conversion should be implemented and executed during calibration. Not afterwards. Doing this extra step could help filter out possible interference. Another area of improvement is the surface emissivity model. In this thesis, surface emissivity is uniformly generated to fall within a given range. A model to include location, time of year, and frequencies to use could help improve temperature retrievals. Also a new LLSE can be developed to include cloudy days or include more instrument noise. Finally, the temperature profile retrievals appear to exhibit excessive noise at times, and the source of this noise should be sought and eliminated.

Appendix A : Select MATLAB Code

A.1 NAST-M Calibration Routine with Spillover Correction.

Important scripts that were used in preparing NAST-M data for temperature retrievals can be found here. All other functions can be located in my home directory.

```
% cnastm
%
% The following script reads in NAST-M data, calibrates, and stores the results to a .mat
file.
% Location of NAST-M data will determine where the data will be stored. For example:
%
% The basic NAST-M data directories will consist of the following:
%
% (mission)/(date)/data
%             /images
%             /log
%
% As NAST-M data is calibrated and ready for saving, a directory called "matlab" is
created at
% the same level and then saved as "(date).mat".
%
% This routine include spillover corrections.
%
% Andrew Sanchez
% 11/10/02

[UIF, UIP] = pickdata('Where is the Raddata Data?', '*raddata')

if (isempty(UIF) == 1 | isempty(UIP) == 1) error('No raddata has been picked.'); end;

fprintf('\nLoading Scans ...\n');
[x,t,rtdf,nav,navt,LL,rtdhead,rtd,tspot,rtdH,rtdA]=load_scans_M(UIF,UIP);

% Load spillover percentage on calibration loads
% eta_A [8x2]
% eta_H [8x2]
load ETA_RESULTS_082401.mat

eta_h=[1-sum(eta_H');eta_H'];
eta_a=[1-sum(eta_A');eta_A'];
```

```
% intialize weight matrix dimensions
% W1 is hot load in this calibration
W1=zeros(17,30); % 17 channels and 29+1 RTDs and corruption
W2=W1; % W2 is ambient load

% load calculated weights of hot load RTD's for Mar. 29, 1999 flight
load hot_load_weights.mat

for i=1:17,
   W1(i,rtdH(:,1))=wgth';
end

w1=[3 4];

% set cold load rtd weights
WgtC=[0.2;0.2;0.2;0.2;0.2];

% Find cold load weights
for i=1:17,
   W2(i,rtdA(:,1))=wgta';
end

% set which spots are ambient load
w2=[24 25];

rtdW.H=rtdH;
rtdW.A=rtdA;

fprintf('\nCalibrating preliminary brightness tempuratures ...\n')
Tb_prelim = calibnastm2(x, rtdf, t, tspot, W1, W2, w1, w2);

W1a=zeros(17,31);
W2a=W1a;

WgtH2=[eta_h [ones(1,9);zeros(2,9)]];  %[3x17]

for i=1:17,
   W1a(i,rtdH(:,1))=WgtH2(1,i).*wgth';
end

W1a(:,30)=WgtH2(2,:)';
W1a(:,31)=WgtH2(3,:)';
w1a=[3 4];

% form vector for 54 and 118
WgtC2=[eta_a [ones(1,9);zeros(2,9)]]; %[3x17]
```

```
% Find cold load weights
for i=1:17,
   W2a(i,rtdA(:,1))=WgtC2(1,i).*wgta';
end

W2a(:,30)=WgtC2(2,:)';
W2a(:,31)=WgtC2(3,:)';
w2a=[24 25];

fprintf('Calibrating brightness tempurature ...\n')
Tb=calibnastm_spillcor(x, rtdf, t, tspot, W1a, W2a, w1a, w2a, Tb_prelim);


if ispc foward_back_slash = '\'; else foward_back_slash = '/'; end;

[status,msg] = mkdir(UIP([1:length(UIP)-5]), 'matlab');

save_path = strcat(UIP([1:length(UIP)-5]), 'matlab', foward_back_slash);

save_name = strcat(UIF(1,[1:7]),'.mat');

fprintf('Saving workspace to %s%s ...', save_path, save_name)

save(strcat(save_path,save_name),'x','Tb','rtdf','t','tspot')

fprintf(' Done.\n\n')
clear
```

A.2 Routine for Interpolating NAST-M Scan Times to GPS Time.

```
% inastm
% Interpolate NASTM GPS data and Proteus Navigational data
% to NASTM t.
%
% The basic NAST-M data directories will consist of the
% following:
%
% (mission)/(date)/data
%              /images
%              /log
%              /gps
%              /proteus
%
% NAST-M gps data need to be read into MATLAB and saved into
```

```
% a folder called "gps". The file should be called
% "(data)_gps.mat". Proteus navigational data should be
% available before running this script as well. Although
% Proteus data is not required at this point, it is
% recommended do have in MATLAB format for retrieval use.
% If it is available, it needs to be saved as "(date)_pro.mat"
% in the proteus folder.
%
% Once all the additional data is interpolated, it is saved
% in the matlab folder as "(date).mat"
%
% Andrew Sanchez
% 11/10/02

% Load Calibrated NASTM data (x, Tb, rtdf, t, tspots)
[nastm_data_name, nastm_data_path] = pickdata('Where is the Reduced Data?', '*mat');

if (isempty(nastm_data_name) == 1) error('No NASTM data has been picked.'); end;

fprintf('\n Loading %s ...',nastm_data_name);
load(strcat(nastm_data_path, nastm_data_name));
fprintf(' Done.\n');

if ispc foward_back_slash = '\'; else foward_back_slash = '/'; end;

% Load raw NASTM gps data (gps_raw)
gps_data_path = strcat(nastm_data_path([1:(length(nastm_data_path)-
7)]),'gps',foward_back_slash)
file_info = dir(fullfile(gps_data_path, '*gps.mat'));
gps_data_name = strvcat(file_info.name);

fprintf('\n Loading %s ...',gps_data_name);
load(strcat(gps_data_path, gps_data_name));
fprintf(' Done.\n');

[y1, m1, d1] = prev_sunday(nastm_data_name([1:7]));

% Converts gps_raw time into posix.
gps_raw(:,7) = datebase(m1,d1,y1)+gps_raw(:,7);

% Fixes any offsets that may occur in NASTM data due to wrong clock settings.
if t(1)> gps_raw(1,7)
   t_fix = t(1)-gps_raw(1,7);
   t = t - t_fix;
else
   t_fix = gps_raw(1,7)-t(1);
```

```
   t = t + t_fix;
end;

% Load Proteus navigational data (pro_nav_raw) if it exists.
proteus_data_path = strcat(nastm_data_path([1:(length(nastm_data_path)-
7)]),'proteus',foward_back_slash);
file_info = dir(fullfile(proteus_data_path, '*pro.mat'));

if isempty(file_info) ~= 1
   proteus_data_name = file_info.name
   fprintf('\n Loading %s ...', proteus_data_name);
   load(strcat(proteus_data_path, proteus_data_name));
   fprintf(' Done.\n\n');

   % Checks for 24:59:59 to 00:00:00 turnovers.
   ind = find(pro_nav_raw(:,9)>pro_nav_raw(1,9));

   % Makes time continuous.
   if ind(length(ind))<length(pro_nav_raw)
      pro_nav_raw([ind(length(ind))+1:length(pro_nav_raw)],9) =
24+pro_nav_raw([ind(length(ind))+1:length(pro_nav_raw)],9);
   end;

   dt = datestr(datenum(proteus_data_name), 23);

   m2 = str2num(dt([1:2]));
   d2 = str2num(dt([4:5]));
   y2 = str2num(dt([7:10]));

   % Convert Proteus time column into posix.
   pro_nav_raw(:,9) = datebase(m2,d2,y2)+3600*pro_nav_raw(:,9);

   num_of_gps_col = 9;

else
   fprintf('\n No Proteus data found. \n\n');

   num_of_gps_col = 3;
end;

gps = zeros(length(t), num_of_gps_col);

fprintf('\n Interpolating ...');
gps(:,1) = interp1(gps_raw(:,7),gps_raw(:,3),t);        % NASTM gps latitude
gps(:,2) = interp1(gps_raw(:,7),gps_raw(:,4),t);        % NASTM gps longitude
gps(:,3) = interp1(gps_raw(:,7),gps_raw(:,5),t);        % NASTM gps altitude
```

```
if num_of_gps_col == 9
   gps(:,4) = interp1(pro_nav_raw(:,9),pro_nav_raw(:,1),t);   % Proteus pitch
   gps(:,5) = interp1(pro_nav_raw(:,9),pro_nav_raw(:,2),t);   % Proteus roll
   gps(:,6) = interp1(pro_nav_raw(:,9),pro_nav_raw(:,5),t);   % Proteus gps latitude
   gps(:,7) = interp1(pro_nav_raw(:,9),pro_nav_raw(:,6),t);   % Proteus gps longitude
   gps(:,8) = interp1(pro_nav_raw(:,9),pro_nav_raw(:,12),t);  % Proteus gps altitude
   gps(:,9) = interp1(pro_nav_raw(:,9),pro_nav_raw(:,4),t);   % Proteus static pressure
end;

fprintf(' Done.\n');

save_name = strcat(nastm_data_name([1:7]),'.mat');

fprintf('\n Saving workspace to %s%s ...', nastm_data_path, save_name)
save(strcat(nastm_data_path,save_name),'x','Tb','rtdf','t','tspot','gps')
fprintf(' Done.\n\n');
clear
```

A.3 Quicklook Chart Generation Script

```
% pnastm
% Gui for generating quicklooks.
%
% The basic NAST-M data directories will consist of the
% following:
%
% (mission)/(date)/data
%             /images
%             /log
%             /gps
%             /proteus (optional but recommended)
%             /figure
%
% This script allows you to pick reduced data for particular
% missions and days and plot various figures such as quick
% looks of brightness temperatures for the 54-GHz and 118-GHz.
%
% There is an added feature to save all the quick look plots
% to a folder called "figure".  Comes in real handy for
% producing web pages.
%
% Andrew Sanchez
% 11/10/02
%
```

```
gui_choice_size = [600 80];
question_str = 'Pick something to plot.';
save_str = 'SaVE JPG';
exit_str = 'EXiT';

start_dir = pwd;

scrsz = get(0,'ScreenSize');
scr_wt = scrsz(3);
scr_ht = scrsz(4);

plot_props = {'Name', 'Plot goes here', ...
      'Position', [(scr_wt-800)/2 (scr_ht-600)/2 800 600], ...
      'Resize', 'off', ...
      'NumberTitle', 'off', ...
      'PaperUnits', 'points', ...
      'PaperSize', [800 600], ...
      'InvertHardCopy', 'off', ...
      'Color', [1 1 1], ...
      'PaperPositionMode', 'auto'};

gui_choice_props = {'name', 'What to plot?', ...
      'resize', 'off', ...
      'numbertitle', 'off', ...
      'menubar', 'none', ...
      'createfcn', '', ...
      'closerequestfcn', 'delete(gcbf)', ...
      'position', [75 75 gui_choice_size], ...
      'tag', 'fig_choice_box'};

border = 20;
btn_ht = gui_choice_size(2)-2*border;
btn_wt = 100;
txt_box_ht = (gui_choice_size(2)-2*border)/2;
txt_box_wt = gui_choice_size(1)-2*btn_wt-4*border;
choice_popupmenu_ht = txt_box_ht;
choice_popupmenu_wt = txt_box_wt;

text_box_location =       [border border+choice_popupmenu_ht txt_box_wt txt_box_ht];
choice_popupmenu_location = [border border choice_popupmenu_wt
choice_popupmenu_ht];
save_btn_location =       [2*border+choice_popupmenu_wt border btn_wt btn_ht];
exit_btn_location =       [3*border+choice_popupmenu_wt+btn_wt border btn_wt
btn_ht];
```

```
plot_choices = ['54 GHz Counts Channels 1-4|', ...
          '54 GHz Counts Channels 5-8|', ...
          '118 GHz Counts Channels 1-4|', ...
          '118 GHz Counts Channels 5-9|', ...
          '54 GHz Brightness Temperature All Channels|', ...
          '54 GHz Brightness Temperature Channel 1|', ...
          '54 GHz Brightness Temperature Channel 2|', ...
          '54 GHz Brightness Temperature Channel 3|', ...
          '54 GHz Brightness Temperature Channel 4|', ...
          '54 GHz Brightness Temperature Channel 5|', ...
          '54 GHz Brightness Temperature Channel 6|', ...
          '54 GHz Brightness Temperature Channel 7|', ...
          '54 GHz Brightness Temperature Channel 8|', ...
          '118 GHz Brightness Temperature All Channels|', ...
          '118 GHz Brightness Temperature Channel 1|', ...
          '118 GHz Brightness Temperature Channel 2|', ...
          '118 GHz Brightness Temperature Channel 3|', ...
          '118 GHz Brightness Temperature Channel 4|', ...
          '118 GHz Brightness Temperature Channel 5|', ...
          '118 GHz Brightness Temperature Channel 6|', ...
          '118 GHz Brightness Temperature Channel 7|', ...
          '118 GHz Brightness Temperature Channel 8|', ...
          'NASTM GPS Latitude Longitude Plot|', ...
          'NASTM GPS Time Altitude Plot|', ...
          'Proteus GPS Latitude Longitude Plot|', ...
          'Proteus GPS Time Altitude Plot|', ...
          'NASTM and Proteus GPS Latitude Longitude Plot|', ...
          'NASTM and Proteus GPS Time Altitude Plot'];

plot_save_names = ['_cnts_5414'; ...
            '_cnts_5458'; ...
            '_cnts11814'; ...
            '_cnts11859'; ...
            '_tb_54_all'; ...
            '_tb_54_ch1'; ...
            '_tb_54_ch2'; ...
            '_tb_54_ch3'; ...
            '_tb_54_ch4'; ...
            '_tb_54_ch5'; ...
            '_tb_54_ch6'; ...
            '_tb_54_ch7'; ...
            '_tb_54_ch8'; ...
            '_tb118_all'; ...
            '_tb118_ch1'; ...
            '_tb118_ch2'; ...
            '_tb118_ch3'; ...
```

```
'_tb118_ch4'; ...
'_tb118_ch5'; ...
'_tb118_ch6'; ...
'_tb118_ch7'; ...
'_tb118_ch8'; ...
'_n_lat_lon'; ...
'_n___t_alt'; ...
'_p_lat_lon'; ...
'_p___t_alt'; ...
'_nplat_lon'; ...
'_np__t_alt'];


[nastm_data_name, nastm_data_path] = pickdata('Where is the data to plot?', '*mat');

if (isempty(nastm_data_name) == 1) error('No NASTM data has been picked.'); end;

fprintf('\n Loading %s ...',nastm_data_name);
load(strcat(nastm_data_path, nastm_data_name));
fprintf(' Done.\n\n');

fig_plot = figure(plot_props{:});
fig_gui_choice = figure(gui_choice_props{:});

text_box = uicontrol('style','text', ...
                'string',question_str, ...
                'horizontalalignment','left', ...
                'units','pixels', ...
                'tag','text_box', ...
                'position',text_box_location);

choice_popupmenu = uicontrol('style','popupmenu', ...
                        'string',plot_choices, ...
                        'units','pixels', ...
                        'tag','choice_popupmenu', ...
                        'position',choice_popupmenu_location, ...
                        'callback','plotdata');

save_btn = uicontrol('style','pushbutton', ...
                'string',save_str, ...
                'position',save_btn_location, ...
                'tag','save_btn', ...
                'callback', 'saveplots');

exit_btn = uicontrol('style','pushbutton', ...
                'string',exit_str, ...
```

```
            'position',exit_btn_location, ...
            'tag','exit_btn', ...
            'callback', 'delete(fig_gui_choice);delete(fig_plot);clear;');

handles = guihandles(fig_gui_choice);
guidata(fig_gui_choice,handles);
choice_list = get(handles.choice_popupmenu,'String');
[row col] = size(gps);
[status,msg] = mkdir(nastm_data_path([1:length(nastm_data_path)-7]), 'figures');
figures_path = strcat(nastm_data_path([1:length(nastm_data_path)-7]),'figures');
```

Appendix B : LLSE Preparation and Development

B.1 TIGR Data Set Reader

```
% readSATIGR
% RVL's satigr reader  2/21/00
% Modified by Andrew Sanchez
% uses sat_vap_press(T) for rel humidity calc.

NUM_TITLE  = 25;
NUM_PROFILES = 1761;

[filename, pathname] = pickdata('Where is the satigr info? (satigr.txt)', '*txt')

if (isempty(filename) == 1 | isempty(pathname) == 1) error('satigr information not
picked.'); end;

fprintf('\n Opening satigr information file.\n\n');

fid = fopen([pathname filename]);

% Displays description of satigr.txt file
for index = 1:NUM_TITLE,
        line = fgetl(fid);
        disp(line);
end

fclose(fid);

fprintf('\n Press any key to continue.\n\n');
pause

fprintf('\n Preparing variables.\n\n');

tigr.press = [1013.00; 955.12; 900.33; 848.69; 800.00; 724.78; 651.04; 584.80; 525.00;
471.86; 423.85; 380.73; 341.99; ...
          307.20; 275.95; 247.90; 222.65; 200.00; 161.99; 131.20; 106.27;  86.07;  69.71;
56.46;  45.73;  37.04; ...
           24.79;  16.60;  11.11;  7.43;  4.98;  3.33;  2.23;  1.50;  1.00;  0.55;  0.30;
0.17;  0.09; 0.05];
```

```
tigr.height = std76mb(tigr.press);

fid = fopen([pathname 'satigr.dat']);

for index = 1:NUM_PROFILES
   tigr.profile(index).num = fscanf(fid,'%f',1);
   tigr.profile(index).lat = fscanf(fid,'%f',1);
   tigr.profile(index).lon = fscanf(fid,'%f',1);
   tigr_date = fscanf(fid,'%f',1);

   if (tigr_date < 9999),
      tigr_date_str = num2str(tigr_date);
      if (length(tigr_date_str) == 3), tigr_date_str = [num2str(0) tigr_date_str]; end;
      tigr.profile(index).mon = str2num(tigr_date_str([3 4]));
      tigr.profile(index).yr  = str2num(tigr_date_str([1 2]));
   elseif (tigr_date < 999999 & tigr_date >= 10000);
      tigr_date_str = num2str(tigr_date);
      if (length(tigr_date_str) == 5), tigr_date_str = [num2str(0) tigr_date_str]; end;
      tigr.profile(index).day = str2num(tigr_date_str([1 2]));
      tigr.profile(index).mon = str2num(tigr_date_str([3 4]));
      tigr.profile(index).yr  = str2num(tigr_date_str([5 6]));
   elseif (tigr_date > 1000000)
      tigr_date_str = num2str(tigr_date);
      if (length(tigr_date_str) == 7), tigr_date_str = [num2str(0) tigr_date_str]; end;
      tigr.profile(index).hr  = str2num(tigr_date_str([1 2]));
      tigr.profile(index).day = str2num(tigr_date_str([3 4]));
      tigr.profile(index).mon = str2num(tigr_date_str([5 6]));
      tigr.profile(index).yr  = str2num(tigr_date_str([7 8]));
   else
      tigr.profile(index).hr = [];
      tigr.profile(index).day = [];
      tigr.profile(index).mon = [];
      tigr.profile(index).yr = [];
   end;

   tigr.profile(index).T = flipud(fscanf(fid,'%g',40));
   tigr.profile(index).Ts = fscanf(fid,'%g',1);
   tigr.profile(index).h2o = flipud(fscanf(fid,'%g',40));
   tigr.profile(index).ozo  = flipud(fscanf(fid,'%g',40));
   tigr.profile(index).dh2o = flipud(fscanf(fid,'%g',39));
   tigr.profile(index).sumdn = flipud(fscanf(fid,'%g',39));
   fgetl(fid);

   tigr.profile(index).h2odn = tigr.press.*tigr.profile(index).h2o./(tigr.profile(index).h2o
+ 0.622)*216./tigr.profile(index).T;
   %tigr.profile(index).h2odn = [1e7*tigr.profile(index).dh2o./diff(tigr.height*1000); 0];
```

```
   fprintf('Finished reading profile    %d\n', index);
end

fclose(fid);

fprintf('\n Saving TIGR ensembles ... ');

save('tigr.mat', 'tigr');

fprintf('Done.\n');

clear all
```

B.2 Calculation of Brightness Temperature Components

```
function Tbparts = simTIGRnadirTbparts(T);
%
% Using the TIGR temperature profiles, the upwelling,
% downwelling, and oneway transmittance coeffiecients are
% calculated for each profile for all NAST-M channels.
%
% Andrew Sanchez
% 11/10/02
%

load tigr.mat

if (nargin == 0)
   % Use default half atmosphere.
   T = [tigr.profile.T];
   T = T(1:22,:);
end

% Load passband characteristics of NASTM radiometer
ANGLES_TO_AVERAGE = 11; % For each spot, average over this many pencil beams
SCAN_ANGLES = -2.8; % nadir

LO_54GHz = 45.995; % From 27mar99;
LO_118GHz = 118.7595;

[F54, p54, F118, p118] = nastm_passband_load(LO_54GHz, LO_118GHz);

% Need to average over secants ...
beamwidth = 7.5; % degrees, FWHM
```

```
sigma = beamwidth/2/sqrt(2*log(2)); % std dev value to create FWHM = beamwidth
                        % +/- 4 sigma should account for 99.99 percent of power
                        % 4 sigma = 4*beamwidth/2/sqrt(2(log(2))) degrees
sigma4 = 4*sigma;
angles_to_avg = linspace(-sigma4, sigma4, ANGLES_TO_AVERAGE); % average over
angles
d_angle = mean(diff(angles_to_avg));
mts_beam = normpdf(angles_to_avg, 0, sigma);
mts_beam118 = ones(length(F118), 1) * mts_beam;
mts_beam54 = ones(length(F54), 1) * mts_beam;


% Initialize surface parameters and angles within antenna pattern

No_Profiles = 1761;

angle_secant = sec((angles_to_avg + SCAN_ANGLES) / 180 * pi);

for index = 1:No_Profiles,

   fprintf('\n Profile %g\n', index);

   fprintf('\n Simulating 54-GHz nadir parts ... ');
   [tb1_54l,tb2_54l,e1_54l,e2_54l] =
tbarray(T(1:21,index),tigr.press(1:21),tigr.profile(index).h2odn(1:21),zeros(21,1),zeros(2
1,1),angle_secant',angle_secant',F54');
   [dtb1_54,dtb2_54,de1_54,de2_54] =
tbarray(T(21:22,index),tigr.press(21:22),tigr.profile(index).h2odn(21:22),[0;0],[0;0],angl
e_secant',angle_secant',F54');
   [tb1_54u,tb2_54u,e1_54u,e2_54u] =
tbarray(T(22:size(T,1),index),tigr.press(22:size(T,1)),tigr.profile(index).h2odn(22:size(T,
1)),zeros(size(T(22:size(T,1),index))),zeros(size(T(22:size(T,1),index))),angle_secant',an
gle_secant',F54');

   Tbparts.NASTM54(index).TU_1 = p54*trapz(mts_beam54.*tb2_54l',2)*d_angle;
   Tbparts.NASTM54(index).TU_2 = p54*trapz(mts_beam54.*(tb2_54l'.*de1_54' +
dtb2_54'),2)*d_angle;
   Tbparts.NASTM54(index).E_1 = p54*trapz(mts_beam54.*e2_54l',2)*d_angle;
   Tbparts.NASTM54(index).E_2 =
p54*trapz(mts_beam54.*(e2_54l'.*de2_54'),2)*d_angle;
   TD54a = (e1_54l'.*de1_54'.*tb1_54u') + (e1_54l'.*dtb1_54') + tb1_54l' +
((e1_54u'.*e1_54l'.*de1_54').*(Tcosmic(F54)'*ones(1,ANGLES_TO_AVERAGE)));
   Tbparts.NASTM54(index).TD_1 =
p54*trapz(mts_beam54.*e2_54l'.*TD54a,2)*d_angle;
   Tbparts.NASTM54(index).TD_2 =
p54*trapz(mts_beam54.*e2_54l'.*de2_54'.*TD54a,2)*d_angle;
```

```
   fprintf('Done.');

   fprintf('\n Simulating 118-GHz nadir parts ... ');
   [tb1_118l,tb2_118l,e1_118l,e2_118l] =
tbarray(T(1:21,index),tigr.press(1:21),tigr.profile(index).h2odn(1:21),zeros(21,1),zeros(2
1,1),angle_secant',angle_secant',F118');
   [dtb1_118,dtb2_118,de1_118,de2_118] =
tbarray(T(21:22,index),tigr.press(21:22),tigr.profile(index).h2odn(21:22),[0;0],[0;0],angl
e_secant',angle_secant',F118');
   [tb1_118u,tb2_118u,e1_118u,e2_118u] =
tbarray(T(22:size(T,1),index),tigr.press(22:size(T,1)),tigr.profile(index).h2odn(22:size(T,
1)),zeros(size(T(22:size(T,1),index))),zeros(size(T(22:size(T,1),index))),angle_secant',an
gle_secant',F118');

   Tbparts.NASTM118(index).TU_1 = p118 * trapz(mts_beam118 .* tb2_118l',2) *
d_angle;
   Tbparts.NASTM118(index).TU_2 = p118 *
trapz(mts_beam118.*(tb2_118l'.*de2_118'+dtb2_118'),2)*d_angle;
   Tbparts.NASTM118(index).E_1 = p118 * trapz(mts_beam118 .* e2_118l',2) *
d_angle;
   Tbparts.NASTM118(index).E_2 = p118 *
trapz(mts_beam118.*(e2_118l'.*de2_118'),2)*d_angle;
   TD118a = (e1_118l'.*de1_118'.*tb1_118u') + (e1_118l'.*dtb1_118') + tb1_118l' +
((e1_118u'.*e1_118l'.*de1_118').*(Tcosmic(F118)'*ones(1,ANGLES_TO_AVERAGE))
);
   Tbparts.NASTM118(index).TD_1 = p118 *
trapz(mts_beam118.*e2_118l'.*TD118a,2) * d_angle;
   Tbparts.NASTM118(index).TD_2 = p118 *
trapz(mts_beam118.*e2_118l'.*de2_118'.*TD118a,2) * d_angle;
   fprintf('Done.\n');
end

return;

fprintf('\n Saving TIGR nadir brightness temperature parts ... ')

save('Tbparts22.mat', 'Tbparts');

fprintf('Done.\n');
```

B.3 ARM Brightness Temperature Components Simulation.

```
%
% Read ARM SONDE file, find usable sondes, and calculate brightness
% temperature components.
```

```
%
% Variable     Name                        Units   SGP
%                                        Min     Max     Delta
% base_time    seconds since epoch          sec     0.0     ----    1
% time_offset  seconds since base_time      sec     0.0     ----    1
% pres         pressure                    hPa     0.0     1100.0  10.0
% tdry         dry bulb temperature        C       -80.0   50.0    10.0
% dp           dewpoint temperature        C       -110.0  50.0    ----
% wspd         wind speed                  m/s     0.0     75.0    ----
% deg          wind direction              deg     0.0     360.0   ----
% rh           relative humidity           pct     0.0     100.0   ----
% u_wind       eastward wind component     m/s     -100.0  100.0   ----
% v_wind       northward wind component    m/s     -100.0  100.0   ----
% wstat        wind status                 none    0.0     ----    ----
% asc          ascent rate                 m/s     -10.0   20.0    5.0
% lat          latitude of sonde launch    deg     -90     90      0.0
% lon          longitude of sonde launch   deg     -180    180     0.0
% alt          meters above sea level      m       0       ----

[nastm_data_name, nastm_data_path] = pickdata('Where is the Reduced Data?', '*mat');

if (isempty(nastm_data_name) == 1) error('No NASTM data has been picked.'); end;

files = dir(nastm_data_path);

if sum(strcmp([nastm_data_name(1:length(nastm_data_name)-4) '_ARM.txt'],
cellstr(strvcat(files.name)))),
   delete([nastm_data_path nastm_data_name(1:length(nastm_data_name)-4)
'_ARM.txt']);
end;

diary([nastm_data_path nastm_data_name(1:length(nastm_data_name)-4) '_ARM.txt']);

fprintf('\nLoading %s ... ', nastm_data_name);
load([nastm_data_path nastm_data_name]);
fprintf('Done.\n');

nastm_start = t(1);
nastm_end = t(length(t));

if ispc foward_back_slash = '\'; else foward_back_slash = '/'; end;

sonde_data_path = strcat(nastm_data_path([1:(length(nastm_data_path)-7)]), 'ARM',
foward_back_slash, 'sondes', foward_back_slash);
sonde_file_info = dir(fullfile(sonde_data_path, '*.cdf'));
sonde_data_list = strvcat(sonde_file_info.name);
```

```
if size(sonde_data_list,1) == 0,
   fprintf('\nCouldn"t find any sondes.');
   diary off;
   break;
else
   fprintf('\nFound %g ARM SONDE(S) for possible use.\n', size(sonde_data_list,1));
end;

use_sonde = zeros(size(sonde_data_list,1),1);
ARMsonde_index = 1;

for sonde_index = 1:size(sonde_data_list,1),

   sonde = readARMsonde(sonde_data_path, sonde_data_list(sonde_index,:));

   if ~isempty(sonde),
      if ((sonde.tsonde(1) >= nastm_start)&(sonde.tsonde(1) <= nastm_end)),

         pt1 = [gps(:,1) gps(:,2)];
         pt2 = ones(length(gps),1)*[sonde.lat(1) sonde.lon(1)];

         distrhkm = deg2km(distance('rh', pt1, pt2));

         if (sum(distrhkm<50)>0),
            fprintf('\n   Sonde is suitable for use.\n');
            use_sonde(sonde_index) = 1;
            ARM.profile(ARMsonde_index).file = sonde_data_list(sonde_index,:);
            ARM.profile(ARMsonde_index).tempi = sonde.tempi;
            ARM.profile(ARMsonde_index).vapi = sonde.vapi;
            ARM.profile(ARMsonde_index).Ts = sonde.Ts;
            ARM.profile(ARMsonde_index).lat = sonde.lat(1);
            ARM.profile(ARMsonde_index).lon = sonde.lon(1);
            ARM.profile(ARMsonde_index).Tbparts = findARMsondeTbparts(sonde);
            ARMsonde_index = ARMsonde_index + 1;
         else
            fprintf('\n   Sonde is too far from aircraft trajectory.\n');
         end;

      else
         fprintf('\n   Sonde time was not in the range of mission duration.\n');
      end;
   else
      fprintf('\n   Sonde failed to have the first 22 TIGR pressure levels\n');
   end
```

```
  clear sonde
end;

sonde_use_list = sonde_data_list(find(use_sonde),:);
fprintf('\nFound %g sonde(s) to use for %s.\n', size(sonde_use_list, 1),
nastm_data_name);
sonde_use_list

fprintf('\nSaving ARM SONDE NAST-M Tb parts to %s ... ', [nastm_data_path
nastm_data_name(1:length(nastm_data_name)-4) '_ARM.mat'])
save([nastm_data_path nastm_data_name(1:length(nastm_data_name)-4) '_ARM.mat'],
'ARM');
fprintf('Done.\n')

fprintf('\nClosing %s file.\n',[nastm_data_path
nastm_data_name(1:length(nastm_data_name)-4) '_ARM.txt']);
diary off;
```

B.4 NAST-M Brightness Temperature Simulation

```
function Tbsim = simTIGRTb(Tbparts, Ts, emis54, emis118, pres);

% Tbsim = simTIGRTb(Tbparts, Ts, emis54, emis118, pres)
%
% simTIGRTb Simulates brightness temperatures that NAST-M would observe.
%
%   Two ways of using simTIGRTb:
%     1. There are no variables passed to simTIGRTb.
%     2. There are 5 variables passed to simTIGRTb.
%
%   Using method 1 will simulate a new realization of brightness
%   temperatures from the TIGR profile ensemble.  Tbsim will look
%   something like this:
%
%     Tbsim.Tb is a 17 x 1761 matrix of NAST-M brightness temperatures.
%
%     Tbsim.param will contain structure fields of emis54, emis118,
%     pres, Ts, Tsemis54, and Tsemis118.  The last two fields are the
%     product of Ts and emis.
%
%   Using method 2 will require all the function inputs to be present.
%   The structure fields of Tbsim.param will be made up of the last 4
%   function inputs.  Tbparts will most likely be generated from a
%   retrieved set of temperatures and emissivity i.e. the ouputs of a
%   MFNN or LLSE.
```

```
%
%   Not having 5 variables passed will result in error.
%
%   Andrew Sanchez
%   10-30-02

load tigr.mat;

if (nargin == 0),
   load Tbparts22.mat;

   Tbsim.param.emis54 = unifrnd(0.95, 1, size(tigr.profile)) +
normrnd(0,0.01,size(tigr.profile));
   Tbsim.param.emis54(find(Tbsim.param.emis54>1)) = 1;
   Tbsim.param.emis54(find(Tbsim.param.emis54<0.95)) = 0.95;

   Tbsim.param.emis118 = (1 + 2*Tbsim.param.emis54)/3 +
normrnd(0,0.01,size(Tbsim.param.emis54));
   Tbsim.param.emis118(find(Tbsim.param.emis118>1)) = 1;
   Tbsim.param.emis118(find(Tbsim.param.emis118<0.9667)) = 0.9667;

   Tbsim.param.pres = unifrnd(std76km(17), std76km(15), size(tigr.profile));
   Tbsim.param.Ts = [tigr.profile.Ts] + normrnd(0,3,size(tigr.profile));

elseif (nargin == 5),
   Tbsim.param.emis54 = emis54;
   Tbsim.param.emis118 = emis118;
   Tbsim.param.pres = pres;
   Tbsim.param.Ts = Ts;
else
   error('Not enough function inputs');
end

Tbsim.param.Tsemis54 = Tbsim.param.Ts.*Tbsim.param.emis54;
Tbsim.param.Tsemis118 = Tbsim.param.Ts.*Tbsim.param.emis118;

Tbsim.Tb = [];

Tb54_1 = [Tbparts.NASTM54.TU_1] ...
    +
[Tbparts.NASTM54.E_1].*(ones(8,1)*([Tbsim.param.emis54].*[Tbsim.param.Ts])) ...
    + [Tbparts.NASTM54.TD_1].*(1 - ones(8,1)*[Tbsim.param.emis54]);

Tb54_2 = [Tbparts.NASTM54.TU_2] ...
    +
[Tbparts.NASTM54.E_2].*(ones(8,1)*([Tbsim.param.emis54].*[Tbsim.param.Ts])) ...
```

```
    + [Tbparts.NASTM54.TD_2].*(1 - ones(8,1)*[Tbsim.param.emis54]);

G54 = (Tb54_2 - Tb54_1)./(log10(tigr.press(22)) - log10(tigr.press(21)));
B54 = Tb54_1 - G54.*log10(tigr.press(21));

Tbsim54 = G54.*(ones(8,1)*log10([Tbsim.param.pres])) + B54;


Tb118_1 = [Tbparts.NASTM118.TU_1] ...
    +
[Tbparts.NASTM118.E_1].*(ones(9,1)*([Tbsim.param.emis118].*[Tbsim.param.Ts])) ...
    + [Tbparts.NASTM118.TD_1].*(1 - ones(9,1)*[Tbsim.param.emis118]);

Tb118_2 = [Tbparts.NASTM118.TU_2] ...
    +
[Tbparts.NASTM118.E_2].*(ones(9,1)*([Tbsim.param.emis118].*[Tbsim.param.Ts])) ...
    + [Tbparts.NASTM118.TD_2].*(1 - ones(9,1)*[Tbsim.param.emis118]);

G118 = (Tb118_2 - Tb118_1)./(log10(tigr.press(22)) - log10(tigr.press(21)));
B118 = Tb118_1 - G118.*log10(tigr.press(21));

Tbsim118 = G118.*(ones(9,1)*log10([Tbsim.param.pres])) + B118;

Tbsim.Tb = [Tbsim54;Tbsim118];

return;
```

B.5 LLSE Development

```
% trainTIGRllse54
%
% Trains an LLSE retrieval algorithm using the TIGR data set.
% Only brightness temperatures from the 54-GHz system are used in
% to develop an estimator.
%
% Andrew Sanchez
% 1/10/03

load tigr.mat

load NASTM_noise_std.mat

Tbsim = simTIGRTbbig;
Tbsim.Tb = Tbsim.Tb(1:8,:); % Use only 54-GHz Tb's.
```

```
[Tbsim.Tbn, Tbsim.Tbmean, Tbsim.Tbstd] = prestd(Tbsim.Tb);
Tbsim.Tbzero = diag(Tbsim.Tbstd)*Tbsim.Tbn;

[Tbsim.param.emis54norm, Tbsim.param.emis54mean, Tbsim.param.emis54std] =
prestd(Tbsim.param.emis54);
Tbsim.param.emis54zero = Tbsim.param.emis54std*Tbsim.param.emis54norm;

%[Tbsim.param.emis118norm, Tbsim.param.emis118mean, Tbsim.param.emis118std] =
prestd(Tbsim.param.emis118);
%Tbsim.param.emis118zero = Tbsim.param.emis118std*Tbsim.param.emis118norm;

[Tbsim.param.presnorm, Tbsim.param.presmean, Tbsim.param.presstd] =
prestd(Tbsim.param.pres);
Tbsim.param.preszero = Tbsim.param.presstd*Tbsim.param.presnorm;

[Tbsim.param.Tsnorm, Tbsim.param.Tsmean, Tbsim.param.Tsstd] =
prestd(Tbsim.param.Ts);
Tbsim.param.Tszero = Tbsim.param.Tsstd*Tbsim.param.Tsnorm;

[Tbsim.param.Tsemis54norm, Tbsim.param.Tsemis54mean, Tbsim.param.Tsemis54std]
= prestd(Tbsim.param.Tsemis54);
Tbsim.param.Tsemis54zero = Tbsim.param.Tsemis54std*Tbsim.param.Tsemis54norm;

%[Tbsim.param.Tsemis118norm, Tbsim.param.Tsemis118mean,
Tbsim.param.Tsemis118std] = prestd(Tbsim.param.Tsemis118);
%Tbsim.param.Tsemis118zero =
Tbsim.param.Tsemis118std*Tbsim.param.Tsemis118norm;

x = [Tbsim.Tbzero; Tbsim.param.preszero];
llse.xmean = [Tbsim.Tbmean; Tbsim.param.presmean];
llse.covx = cov(x');

llse.covn = diag([std_NASTM(3,1:8).^2 0]);

Ttigr.T = [];
T = [tigr.profile.T];
T = T(1:22,:);

for index = 1:27,
   Ttigr.T = [Ttigr.T; T];
end

Ttigr.T = reshape(Ttigr.T, 22, 27*1761);

[Ttigr.Tnorm, Ttigr.Tmean, Ttigr.Tstd] = prestd(Ttigr.T);
Ttigr.Tzero = diag(Ttigr.Tstd)*Ttigr.Tnorm;
```

```
y = [Ttigr.Tzero; Tbsim.param.Tszero; Tbsim.param.emis54zero;
Tbsim.param.Tsemis54zero];
llse.ymean = [Ttigr.Tmean; Tbsim.param.Tsmean; Tbsim.param.emis54mean;
Tbsim.param.Tsemis54mean];

llse.covyx = y*x'./(length(x)-1);

%save llse54.mat llse
break


noi = [diag(std_NASTM(3,1:8))*normrnd(0,1,size(Tbsim.Tb));
zeros(1,length(Tbsim.Tb))];

Tllse = diag([Ttigr.Tmean; Tbsim.param.Tsmean; Tbsim.param.emis54mean;
Tbsim.param.Tsemis54mean])*ones(size(y)) + llse.covyx*inv(llse.covx + llse.covn)*(x +
noi);

delta_Tllse_rms = std((Ttigr - Tllse(1:22,:))', 1);

Tshat = Tllse(23,:);
emis54hat = Tllse(24,:);
emis54hat(find(emis54hat>1)) = 1;

emis54_rmse = std(Tbparam.emis54 - emis54hat, 1);
Ts_rmse = std(Tbparam.Ts - Tshat, 1);


%press = [satigr(3,1:22,1)];
%hold on
subplot(2,2,1)
semilogy(delta_Tllse_rms, tigr.press(1:22),'r.-')
set(gca,'YDir','reverse')
axis([0 5 80 1100])
grid on
title('Pressure vs Temperature Error')
xlabel('Error Temperature (K)')
ylabel('Pressure (mb)')
%legend('MFNN', 'Reconst Error', 'A Priori', 'LLSE')

subplot(2,2,2)
plot(Tbparam.Ts - Tshat)
title({'Surface Temperature LLSE retrival', ['RMSE = ' num2str(Ts_rmse) 'K']})
xlabel('Profile Number')
ylabel('Error (K)')
```

```
subplot(2,2,3)
plot(Tbparam.emis54 - emis54hat)
title({'54-GHz Surface Emissivity LLSE retrival', ['RMSE = ' num2str(emis54_rmse)]})
xlabel('Profile Number')
ylabel('Error')
break
subplot(2,2,4)
plot(Tbparam.emis118 - emis118hat)
title({'118-GHz Surface Emissivity LLSE retrival', ['RMSE = '
num2str(emis118_rmse)]})
xlabel('Profile Number')
ylabel('Error')
```

B.6 Temperature Retrieval Code

```
function retTprofile54(s_in)

% Opens NAST-M data for temperature retrieval.
%
% Andrew Sanchez
% 1/28/03

load tigr.mat

load arm.mat;
arm_index = 16;

load gainbase.mat
gain54 = gainbase.mar1700.g54;
base54 = gainbase.mar1700.b54;

load 17Mar00.mat
ddmmmyy = '17Mar00';
%scan_ind = 1009;
scan_ind = [1300 1400 1580 1900 2020 2195 2300 2370];
scan = scan_ind(s_in);
RANGE = scan:scan + 75 - 0;

progps = 0;
useTbcorr = 2;



if progps,
```

```
   Tb = removeTbroll(Tb, gps);
   pres = 1000/14.5*gps(RANGE,9)';
else
   pres = std76km(0.001*gps(RANGE,3))';
end;

M = 4;

Tbp = Tb(:,:,RANGE);
%size(Tbp)
%Tbp(1,:,:)

for index=1:17,
   Tbf(index,:,:) = filtfiltcol(ones(1,M)/M, 1, squeeze(Tbp(index,:,:))')';
end
Tb0 = squeeze(Tb(:,12,RANGE));
%Tb0 = squeeze(Tbz(:,12,:));
Tb0_54 = Tb0(1:8,:) + gain54*Tb0(1:8,:) + base54*ones(size(Tb0(1:8,:)));
Tb0hat = [Tb0_54];

Tb1 = squeeze(Tb(:,13,RANGE));
%Tb1 = squeeze(Tbz(:,13,:));
Tb1_54 = Tb1(1:8,:) + gain54*Tb1(1:8,:) + base54*ones(size(Tb1(1:8,:)));
Tb1hat = [Tb1_54];

Tb2 = squeeze(Tb(:,14,RANGE));
%Tb2 = squeeze(Tbz(:,14,:));
Tb2_54 = Tb2(1:8,:) + gain54*Tb2(1:8,:) + base54*ones(size(Tb2(1:8,:)));
Tb2hat = [Tb2_54];

Tb3 = squeeze(Tb(:,15,RANGE));
%Tb3 = squeeze(Tbz(:,15,:));
Tb3_54 = Tb3(1:8,:) + gain54*Tb3(1:8,:) + base54*ones(size(Tb3(1:8,:)));
Tb3hat = [Tb3_54];

Tb4 = squeeze(Tb(:,16,RANGE));
%Tb4 = squeeze(Tbz(:,16,:));
Tb4_54 = Tb4(1:8,:) + gain54*Tb4(1:8,:) + base54*ones(size(Tb4(1:8,:)));
Tb4hat = [Tb4_54];

Tb5 = squeeze(Tb(:,17,RANGE));
%Tb5 = squeeze(Tbz(:,17,:));
Tb5_54 = Tb5(1:8,:) + gain54*Tb5(1:8,:) + base54*ones(size(Tb5(1:8,:)));
Tb5hat = [Tb5_54];

Tb6 = squeeze(Tb(:,18,RANGE));
```

```
%Tb6 = squeeze(Tbz(:,18,:));
Tb6_54 = Tb6(1:8,:) + gain54*Tb6(1:8,:) + base54*ones(size(Tb6(1:8,:)));
Tb6hat = [Tb6_54];


%Tbsim = simARMTb;
%Tbsim.Tb = Tbsim.Tb(1:14,arm_index);

if useTbcorr,
   fprintf('Using Corrected Brightness Temperatures.\n');
   [T0hat, Ts0hat, emis54_0hat, Tsemis54_0hat] = retTIGRllse54(Tb0hat, pres);
   [T1hat, Ts1hat, emis54_1hat, Tsemis54_1hat] = retTIGRllse54(Tb1hat, pres);
   [T2hat, Ts2hat, emis54_2hat, Tsemis54_2hat] = retTIGRllse54(Tb2hat, pres);
   [T3hat, Ts3hat, emis54_3hat, Tsemis54_3hat] = retTIGRllse54(Tb3hat, pres);
   [T4hat, Ts4hat, emis54_4hat, Tsemis54_4hat] = retTIGRllse54(Tb4hat, pres);
   [T5hat, Ts5hat, emis54_5hat, Tsemis54_5hat] = retTIGRllse54(Tb5hat, pres);
   [T6hat, Ts6hat, emis54_6hat, Tsemis54_6hat] = retTIGRllse54(Tb6hat, pres);
else
   fprintf('Using Uncorrected Brightness Temperatures.\n');
   [T0hat, Ts0hat, emis54_0hat, Tsemis54_0hat] = retTIGRllse54(Tb0(1:8,:), pres);
   [T1hat, Ts1hat, emis54_1hat, Tsemis54_1hat] = retTIGRllse54(Tb1(1:8,:), pres);
   [T2hat, Ts2hat, emis54_2hat, Tsemis54_2hat] = retTIGRllse54(Tb2(1:8,:), pres);
   [T3hat, Ts3hat, emis54_3hat, Tsemis54_3hat] = retTIGRllse54(Tb3(1:8,:), pres);
   [T4hat, Ts4hat, emis54_4hat, Tsemis54_4hat] = retTIGRllse54(Tb4(1:8,:), pres);
   [T5hat, Ts5hat, emis54_5hat, Tsemis54_5hat] = retTIGRllse54(Tb5(1:8,:), pres);
   [T6hat, Ts6hat, emis54_6hat, Tsemis54_6hat] = retTIGRllse54(Tb6(1:8,:), pres);
end;

scrsz = get(0,'ScreenSize');
scr_wt = scrsz(3);
scr_ht = scrsz(4);

figure(5);
clf
subplot(1,2,1)
semilogy(arm.profile(arm_index).tempi, tigr.press(1:22),'r.', mean(T3hat,2),
tigr.press(1:22),'b.')
set(gca,'YDir','reverse')
axis([200 300 80 1100])
grid on
title(['Pressure vs Temperature ' ddmmmyy])
xlabel('Temperature (K)')
ylabel('Pressure (mb)')

subplot(1,2,2)
semilogy(arm.profile(arm_index).tempi - mean(T3hat,2), tigr.press(1:22),'b')
```

```
set(gca,'YDir','reverse')
axis([-10 10 80 1100])
grid on
title(['Pressure vs Temperature Error ' ddmmmyy])
xlabel('Error Temperature (K)')
ylabel('Pressure (mb)')

set(gcf,'Position', [(scr_wt-800)/2 (scr_ht-600)/2 800 600])
print( gcf, '-djpeg', '-r96', [ddmmmyy '_temp_' num2str(RANGE(1))])

T = cat(3, ...
   T0hat', ...
   T1hat', ...
   T2hat', ...
   T3hat', ...
   T4hat', ...
   T5hat', ...
   T6hat');

T = shiftdim(T,1);

M = 4;
MT = 3;

showT
%break
showTb
%showTbdelta
clear all
break
figure

ch = 6;

plot(squeeze(Tbzerof(ch,8:14,:)), squeeze(Tbzerof(ch + 8,8:14,:)),'.')
axis([-1 1 -1 1])
title(['54-GHz Ch ', num2str(ch), ' vs. 118-GHz Ch ', num2str(ch)])
xlabel(['54-GHz Ch ', num2str(ch)])
ylabel(['118-GHz Ch ', num2str(ch)])

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [That, Tshat, emis54hat, Tsemis54hat] = retTIGRllse(Tbs, pres);
% retTIGRllse

load llse54.mat;
```

```
x = [Tbs; pres] - diag(llse.xmean)*ones(size([Tbs; pres]));

%n = sqrt(llse.covn)*normrnd(0,1,size(x));

yhat = diag(llse.ymean)*ones(size(llse.covyx,1), size(x,2)) + llse.covyx*inv(llse.covx +
llse.covn)*(x);% + n);
That = yhat(1:22,:);
Tshat = yhat(23,:);

emis54hat = yhat(24,:);
emis54hat(find(emis54hat>1)) = 1;
emis54hat(find(emis54hat<0.95)) = 0.95;

Tsemis54hat = yhat(25,:);

Return

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% showT(T, pres_index)
%
% Displays 2-D image of retrieved temperatures.


%RANGE([1 length(RANGE)]) = [];
%r = size(T,3);

for index = 1:22,
   Tf(index,:,:) = filtfiltcol(ones(1,MT)/MT,1,squeeze(T(index,:,:))')';
end

Tmean = mean(T,3);
Tmean_m = reshape(Tmean(:)*ones(1,size(T,3)), size(T));
Tzero = Tf - Tmean_m; % 26x9xr

for index = 1 : 22,
   %Tzerof(index,:,:)=Tzero(index,:,:);
   %Tzerof(index,:,:) = filter2((1/9)*ones(3,3), squeeze(Tzero(index,:,:)));
   Tzerof(index,:,:) = wiener2(squeeze(Tzero(index,:,:)), [4 4]);
   %Tzerof(index,:,:) = medfilt2(squeeze(Tzero(index,:,:)), [3 3]);
   %Tzerofc(index,:,:) = Tzerof(index,:,:);
   %Tzerofi(index,:,:) = interp2(squeeze(Tzerofc(index,:,:)),2);
end

sc = 1.5;
```

```
in = 4;

figure(1)
clf
show8stripret(Tzerof([1 11 14 17 19],:,:), [1:5], ':', [], [-sc sc], [in in], 'gray(256)');
%show8stripret(Tzerof([12:22],:,:), [1:11], ':', [], [-sc sc], [in in], 'jet(256)');
titlesub(['NAST-M Retrieval ' ddmmmyy]);
subtitle(sprintf_mat('Altitude %2.4f km, Mean %3.1f K', std76mb(tigr.press([1 11 14 17
19])), mean(Tmean([1 11 14 17 19],:),2) ));
ax=getsub;
set(ax(1), 'XTickLabel', hrminsec(t(RANGE(get(ax(1), 'XTick')))));

scrsz = get(0,'ScreenSize');
scr_wt = scrsz(3);
scr_ht = scrsz(4);

set(gcf,'Position', [(scr_wt-800)/2 (scr_ht-600)/2 800 600])
%print( gcf, '-djpeg', '-r96', [ddmmmyy '_ret54_' num2str(RANGE(1))])

foo1 = squeeze(Tzerof(1,:,:));
foo2 = squeeze(Tzerof(11,:,:));
foo3 = squeeze(Tzerof(14,:,:));
foo4 = squeeze(Tzerof(17,:,:));
foo5 = squeeze(Tzerof(19,:,:));

mean([std(foo1(:)) std(foo2(:)) std(foo3(:)) std(foo4(:)) std(foo5(:))])


if progps,
   figure(4)
   clf
   set(gcf,'Position', [(scr_wt-800)/2 (scr_ht-600)/2 800 600])

   plot(t(RANGE),gps(RANGE,5)+4)
   grid on
   title(['Proteus Roll ' ddmmmyy])
   xlabel('Hour');
   set(gca,'XLim',[t(RANGE(1)) t(RANGE(length(RANGE)))]);
   set(gca,'XTickLabel',hrminsec(get(gca,'XTick')));
   %print( gcf, '-djpeg', '-r96', [ddmmmyy '_roll_' num2str(RANGE(1))])

   figure(6)
   clf
   set(gcf,'Position', [(scr_wt-800)/2 (scr_ht-600)/2 800 600])

   plot(t(RANGE),gps(RANGE,4))
```

```
   grid on
   title(['Proteus Pitch ' ddmmmyy])
   xlabel('Hour');
   set(gca,'XLim',[t(RANGE(1)) t(RANGE(length(RANGE)))]);
   set(gca,'XTickLabel',hrminsec(get(gca,'XTick')));
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Displays NAST-M brightness temperature perturbations from the mean.
%
sc = 1.5;
in = 4;


Tbp = Tb(:,5:23,RANGE);
%Tbp = Tbz(:,5:23,:);

for index=1:17,
   Tbf(index,:,:) = filtfiltcol(ones(1,M)/M, 1, squeeze(Tbp(index,:,:))')';
end



Tbmean = mean(Tbf,3);
Tbmean_m = reshape(Tbmean(:)*ones(1,size(Tbf,3)), size(Tbf)); % 17x25xr
Tbzero = Tbf - Tbmean_m;

for index=1:17,
   %Tb(index,:,:)=filter2((1/9)*ones(3,3), squeeze(Tbp(index,:,:)));
   Tbzerof(index,:,:) = wiener2(squeeze(Tbzero(index,:,:)), [4 4]);
   %Tb(index,:,:)=medfilt2(squeeze(Tbp(index,:,:)), [3 3]);
end


figure(2)
clf
show8strip2(Tbzerof([1 4 5 6 7],:,:), [1:5], ':', [], [-sc sc], [in in], 'gray(256)');
titlesub(['NAST-M 54-GHz RADIOMETER ' ddmmmyy]);
subtitle(sprintf_mat('54-GHz  Ch %d, Mean %3.1f K', [1 4 5 6 7], mean(Tbmean([1 4 5 6
7],:),2) ));
ax=getsub;
set(ax(1), 'XTickLabel', hrminsec(t(RANGE(get(ax(1), 'XTick')))));

scrsz = get(0,'ScreenSize');
scr_wt = scrsz(3);
```

```
scr_ht = scrsz(4);

set(gcf,'Position', [(scr_wt-800)/2 (scr_ht-600)/2 800 600])
print( gcf, '-djpeg', '-r96', [ddmmmyy '_54_' num2str(RANGE(1))])

figure(3)
clf
show8strip2(Tbzerof([9 12 13 14 15],:,:), [1:5], ':', [], [-sc sc], [in in], 'gray(256)');
titlesub(['NAST-M 118-GHz RADIOMETER ' ddmmmyy]);
subtitle(sprintf_mat('118-GHz  Ch %d, Mean %3.1f K', [1 4 5 6 7], mean(Tbmean([9 12
13 14 15],:),2) ));
ax=getsub;
set(ax(1), 'XTickLabel', hrminsec(t(RANGE(get(ax(1), 'XTick')))));

set(gcf,'Position', [(scr_wt-800)/2 (scr_ht-600)/2 800 600])
print( gcf, '-djpeg', '-r96', [ddmmmyy '_118_' num2str(RANGE(1))])
```

Appendix C : More Images

C.1 Additional Imagery from CLOUDIOP

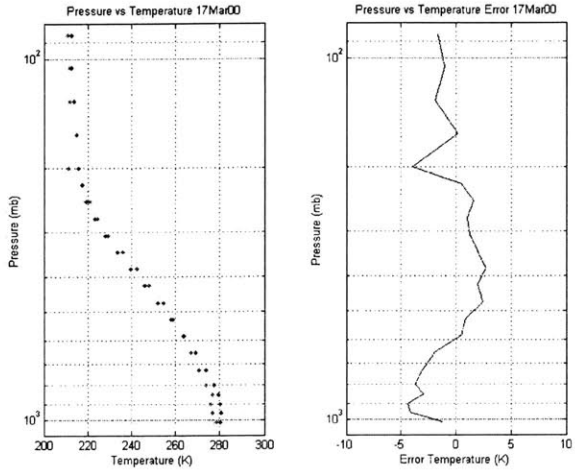


Figure C-1 17Mar00 Temperature Retrieval Comparison for index 1400.

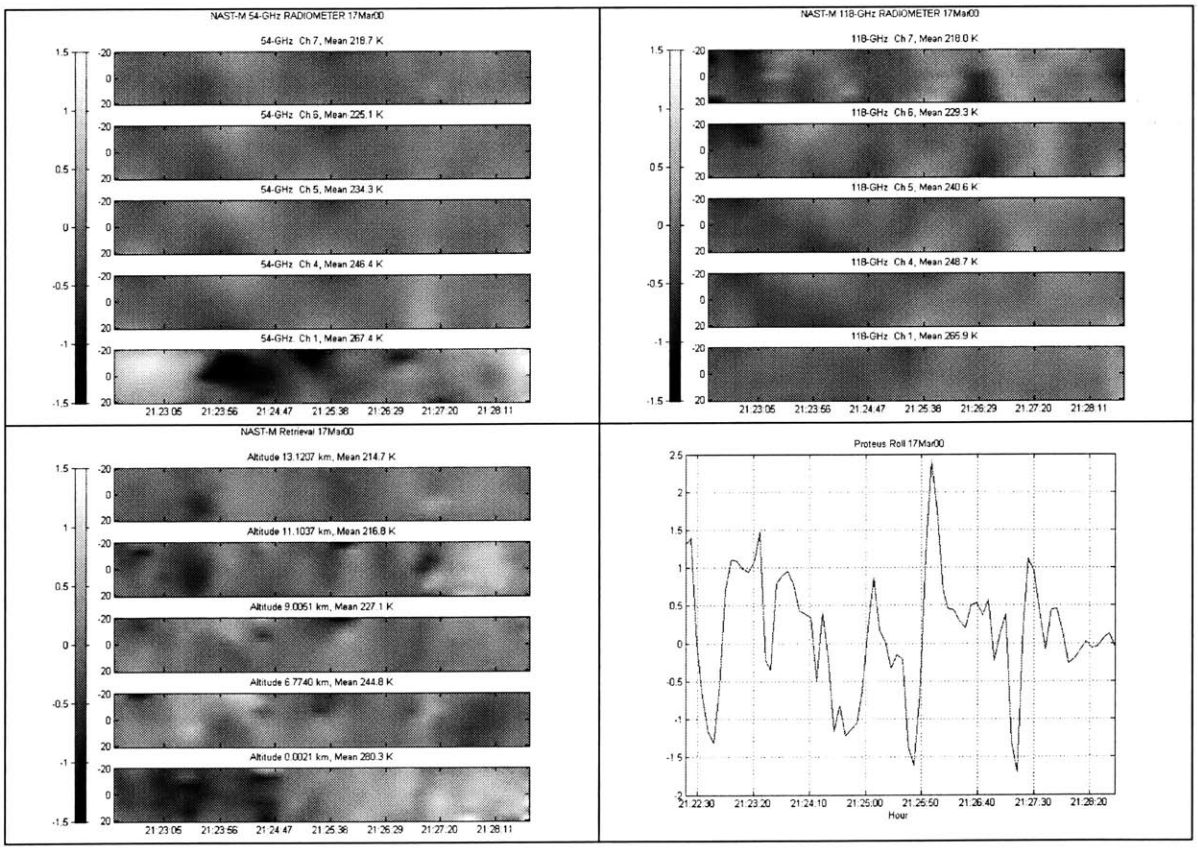


Figure C-2 17Mar00 Temperature Imagery for index 1400.

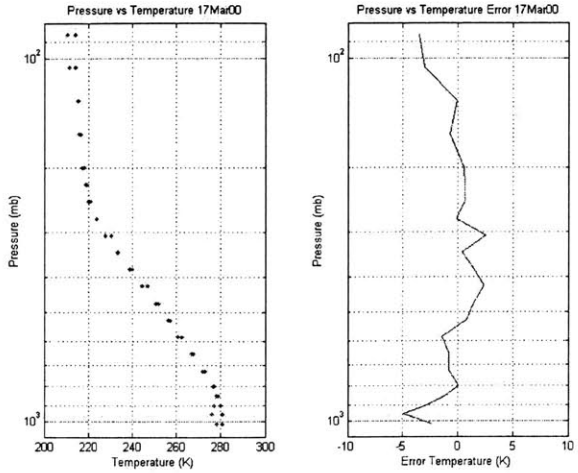


Figure C-3 17Mar00 Temperature Retrieval Comparison for index 1900.

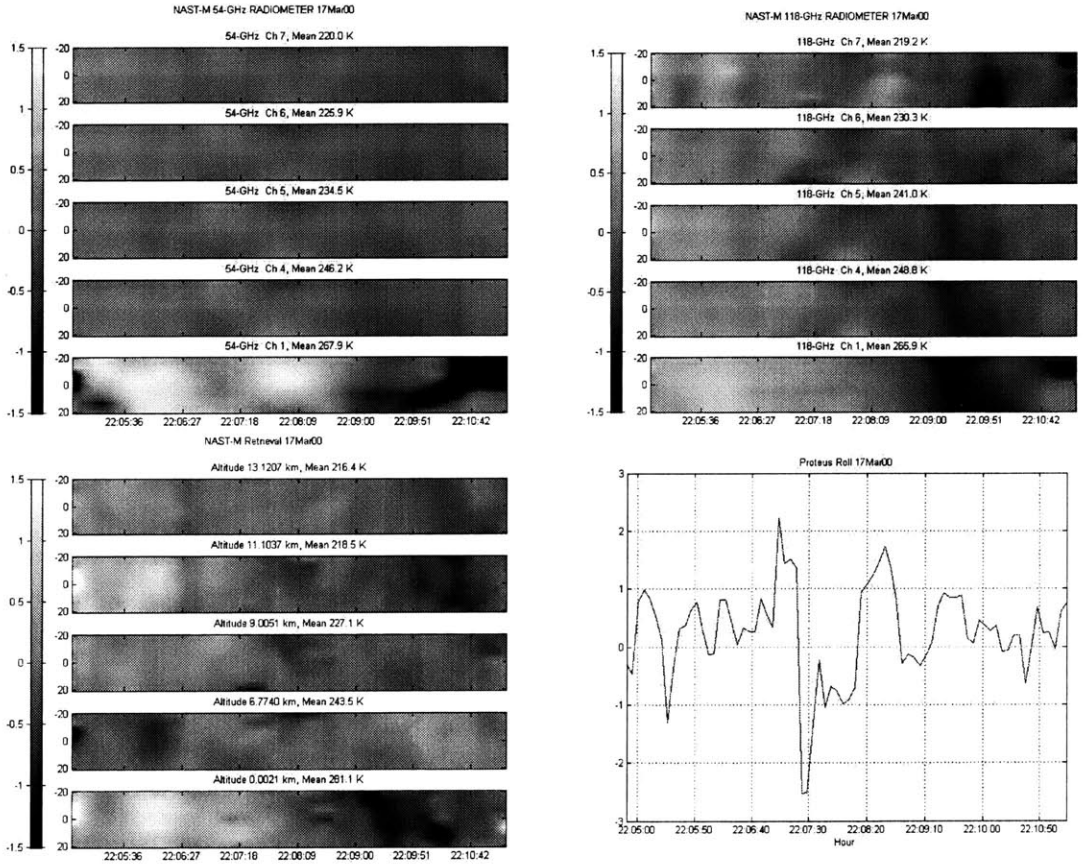


Figure C-4 17Mar00 Temperature Imagery for index 1900.

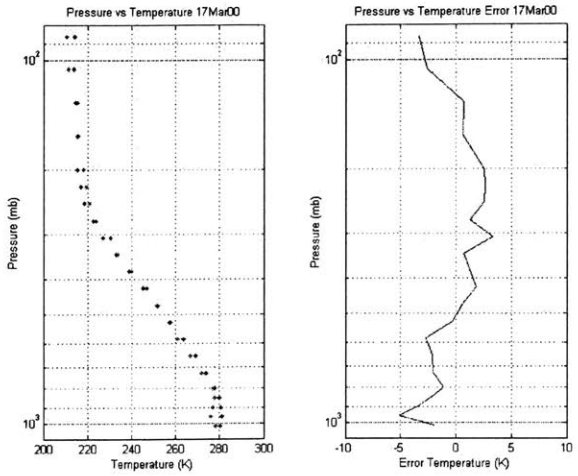


Figure C-5 17Mar00 Temperature Retrieval Comparison for index 2020.

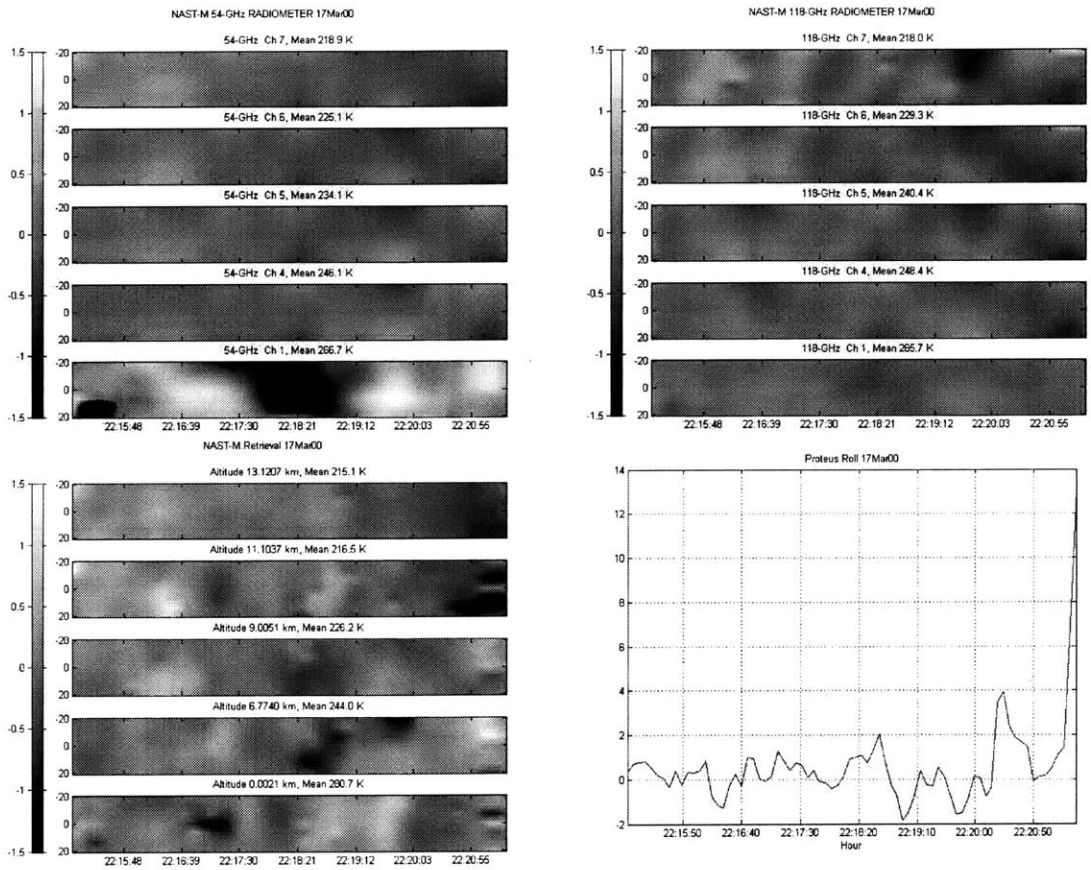


Figure C-6 17Mar00 Temperature Imagery for index 2020.

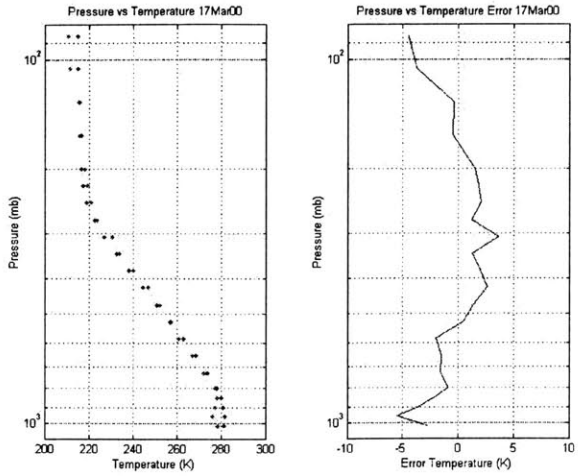


Figure C-7 17Mar00 Temperature Retrieval Comparison for index 2195.

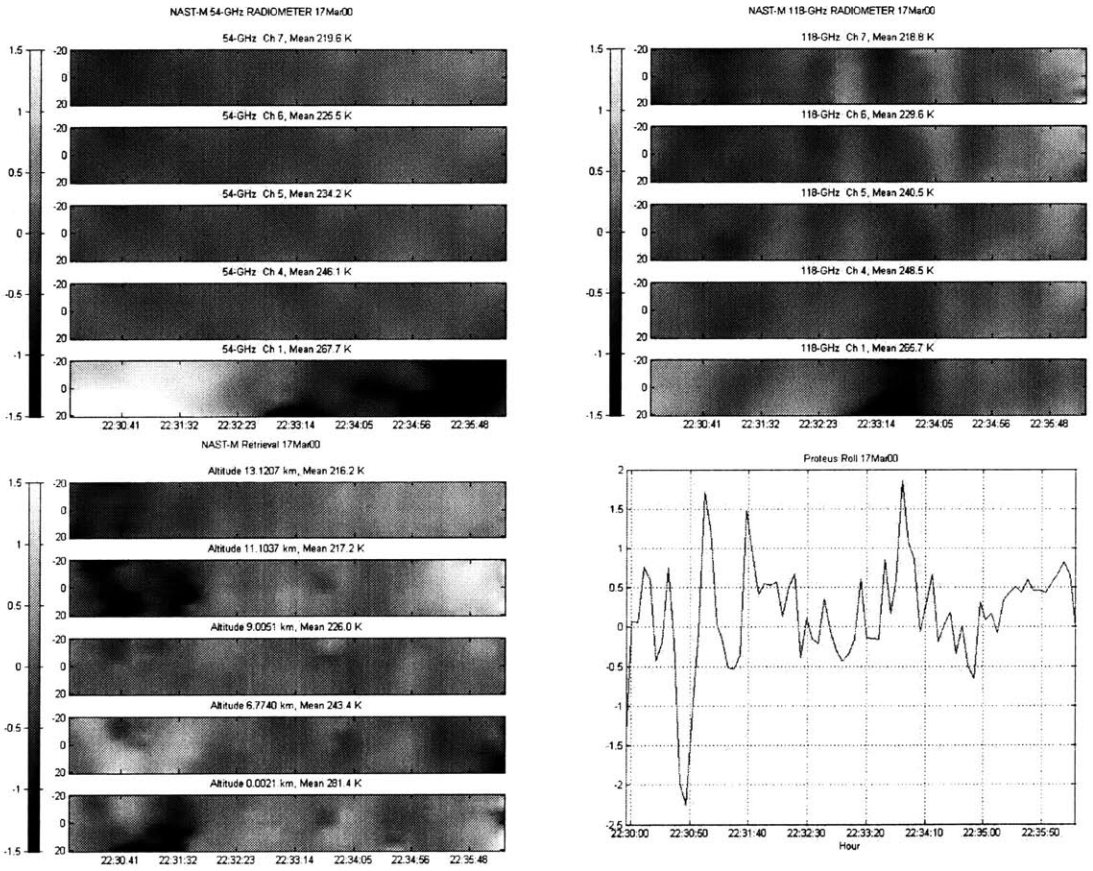


Figure C-8 17Mar00 Temperature Imagery for index 2195.

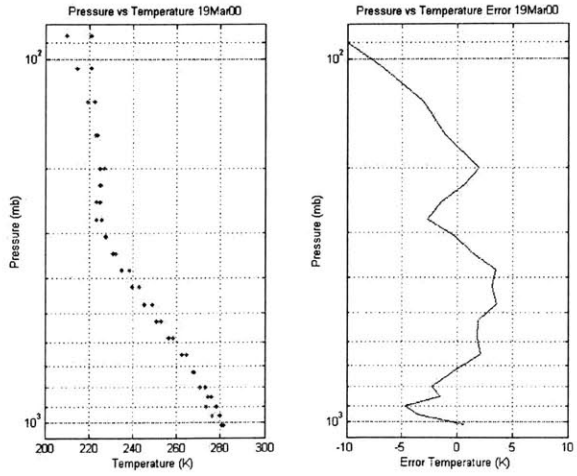


Figure C-9 19Mar00 Temperature Retrieval Comparison for index 1110.

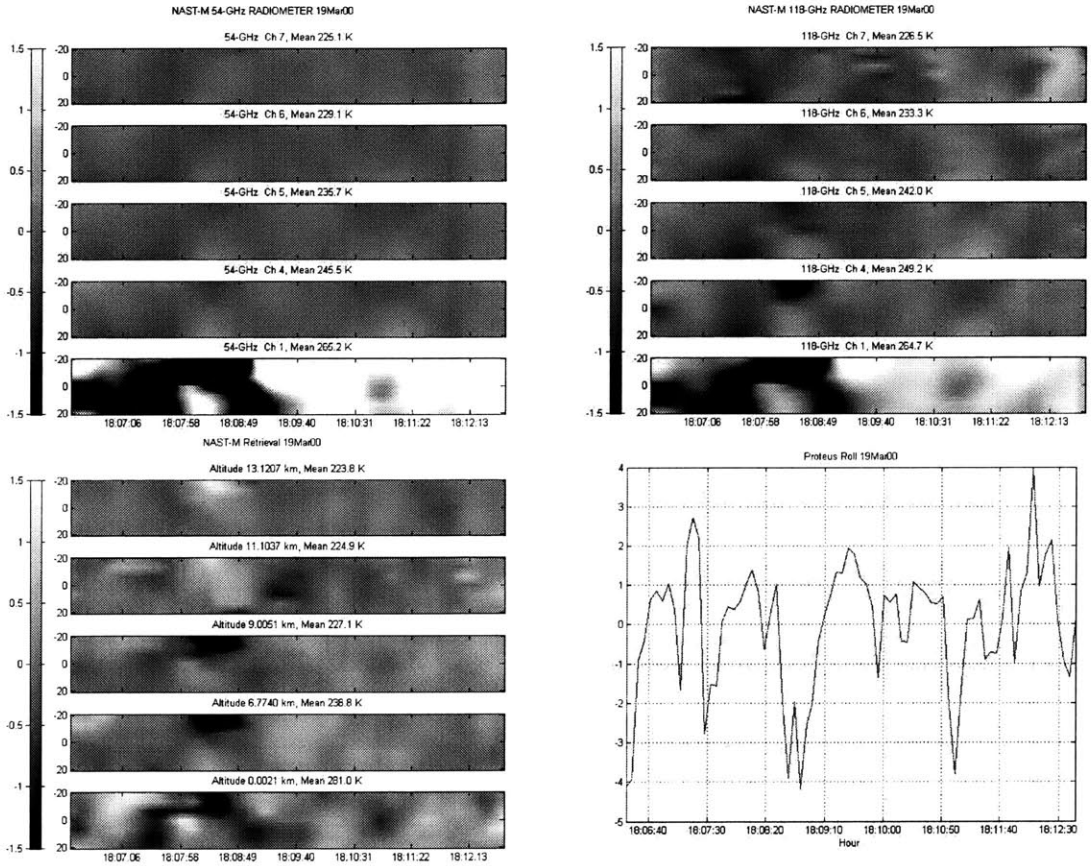


Figure C-10 19Mar00 Temperature Imagery for index 1110.

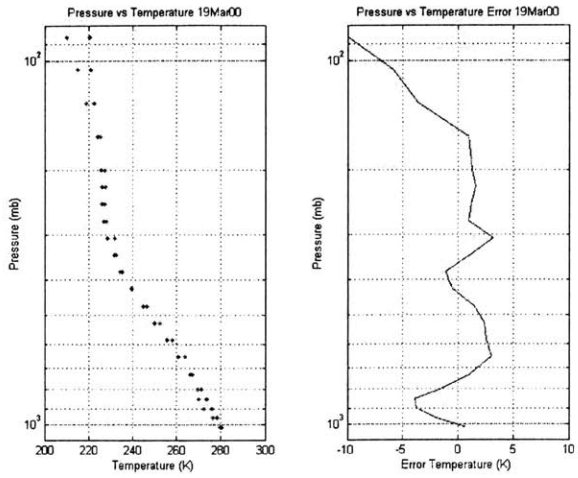


Figure C-11 19Mar00 Temperature Retrieval Comparison for index 1260.

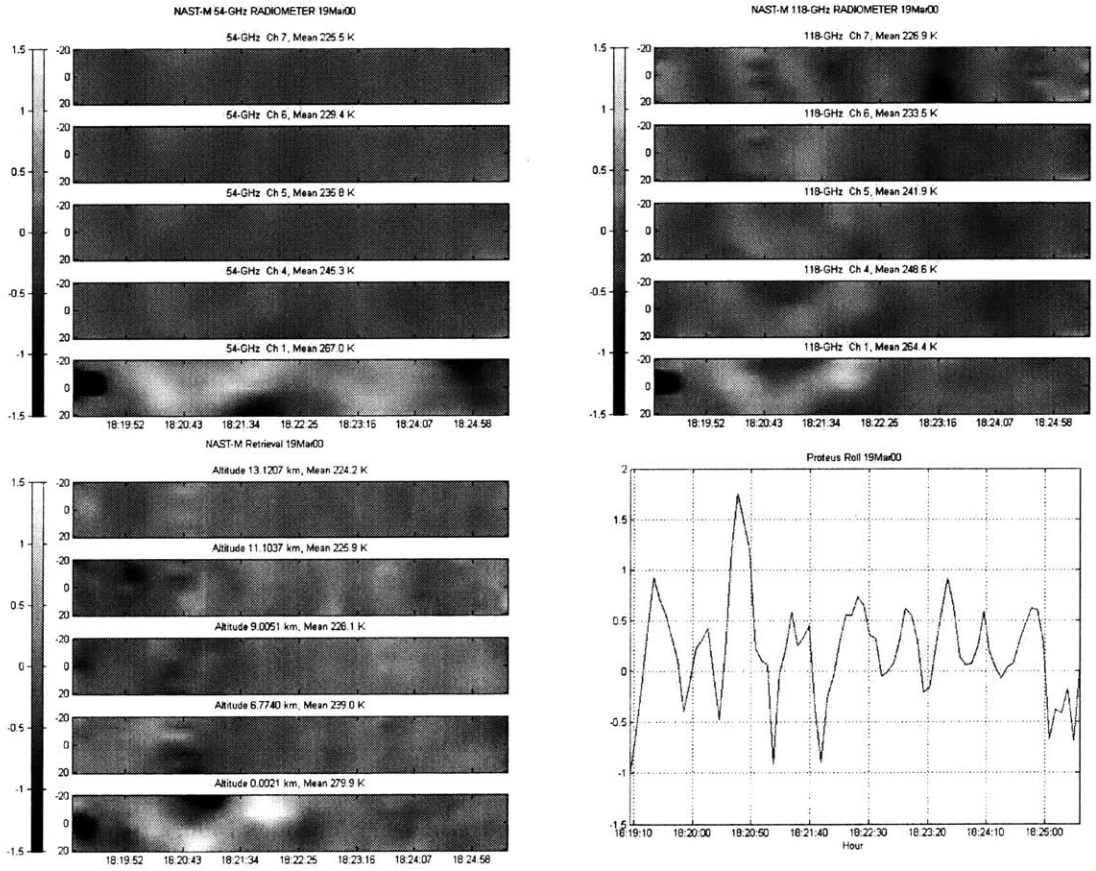


Figure C-12 19Mar00 Temperature Imagery for index 1260.

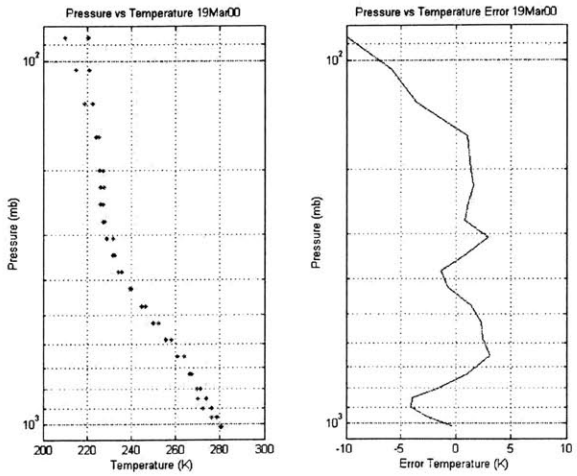


Figure C-13 19Mar00 Temperature Retrieval Comparison for index 1400.

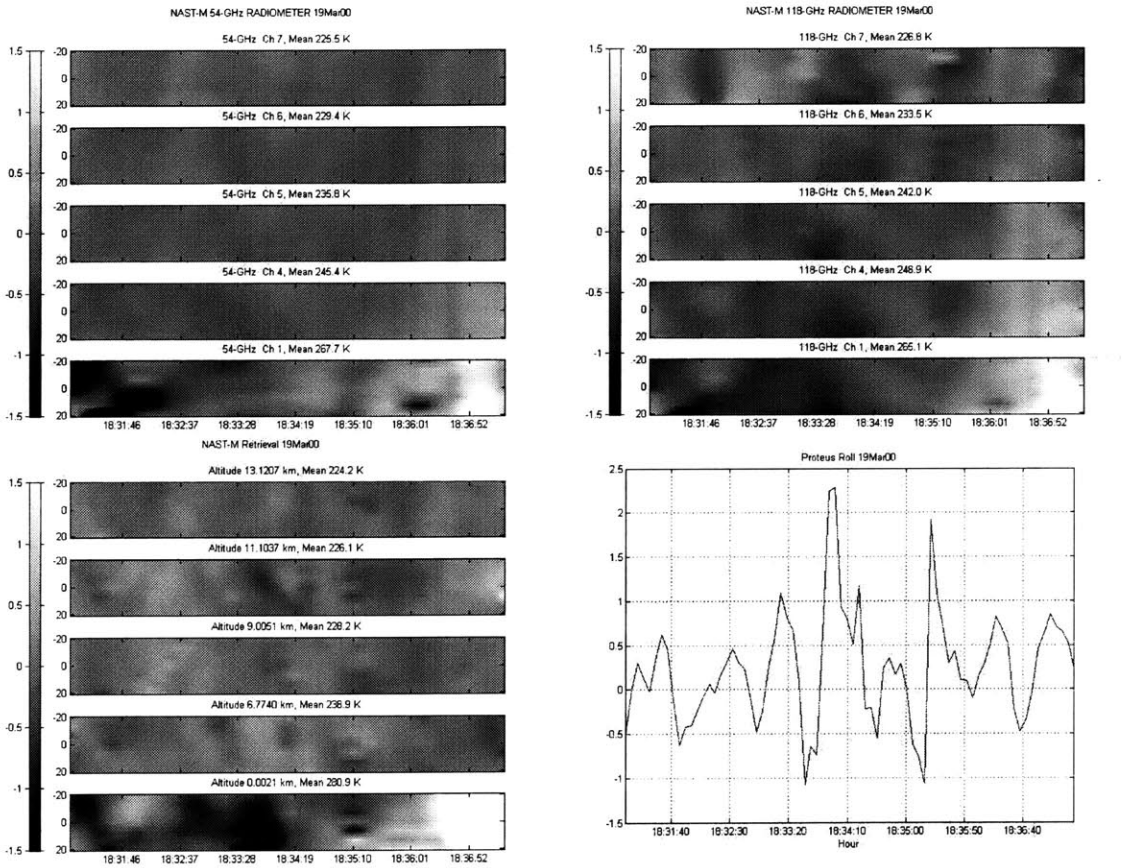


Figure C-14 19Mar00 Temperature Imagery for index 1400.

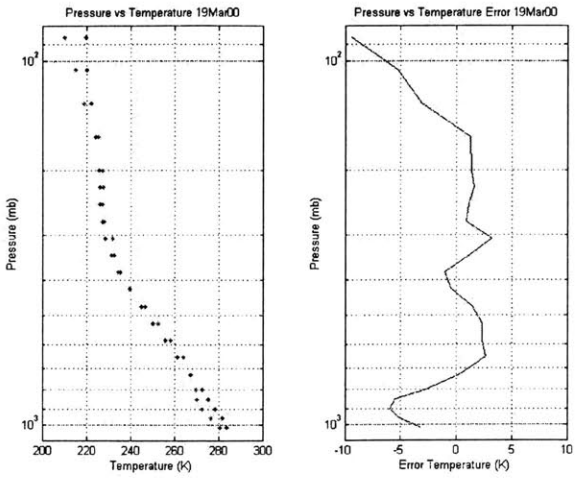


Figure C-15 19Mar00 Temperature Retrieval Comparison for index 1460.

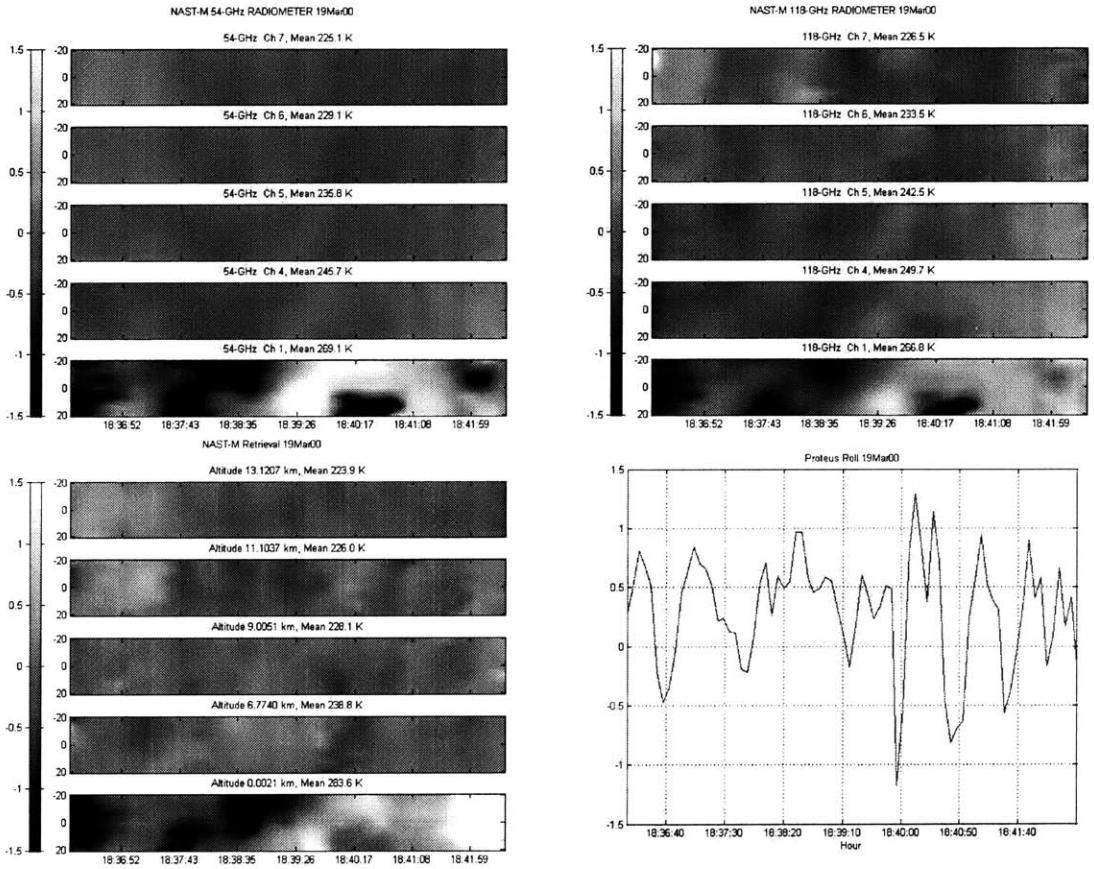


Figure C-16 19Mar00 Temperature Imagery for index 1460.

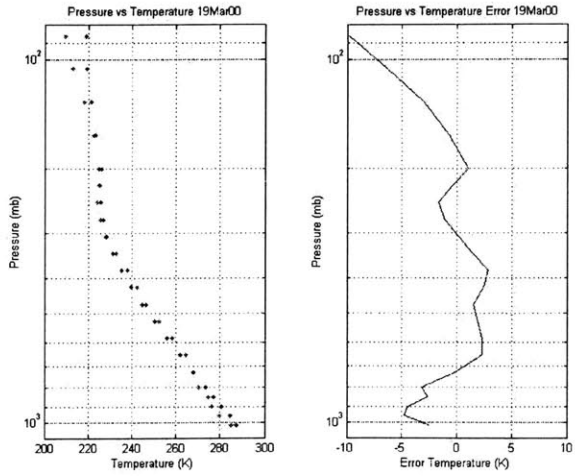


Figure C-17 19Mar00 Temperature Retrieval Comparison for index 1550.

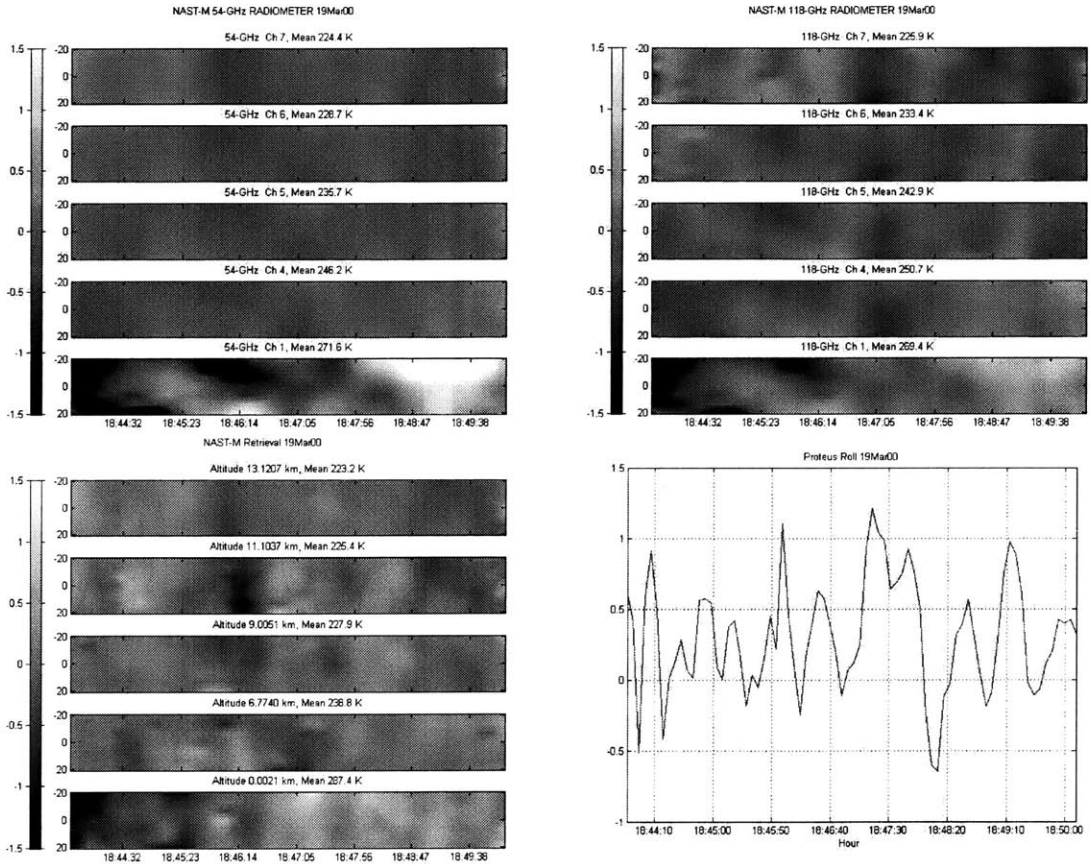


Figure C-18 19Mar00 Temperature Imagery for index 1550

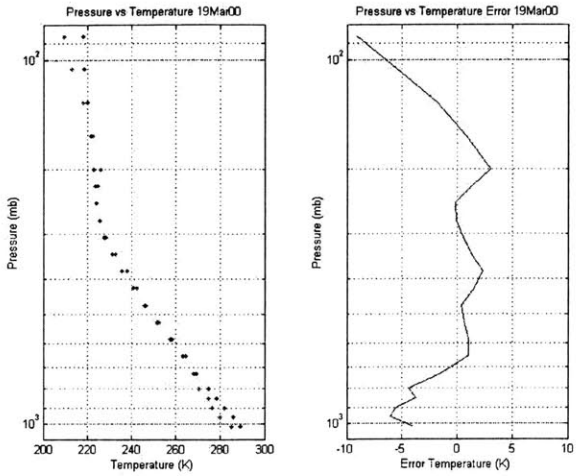


Figure C-19 19Mar00 Temperature Retrieval Comparison for index 1735.

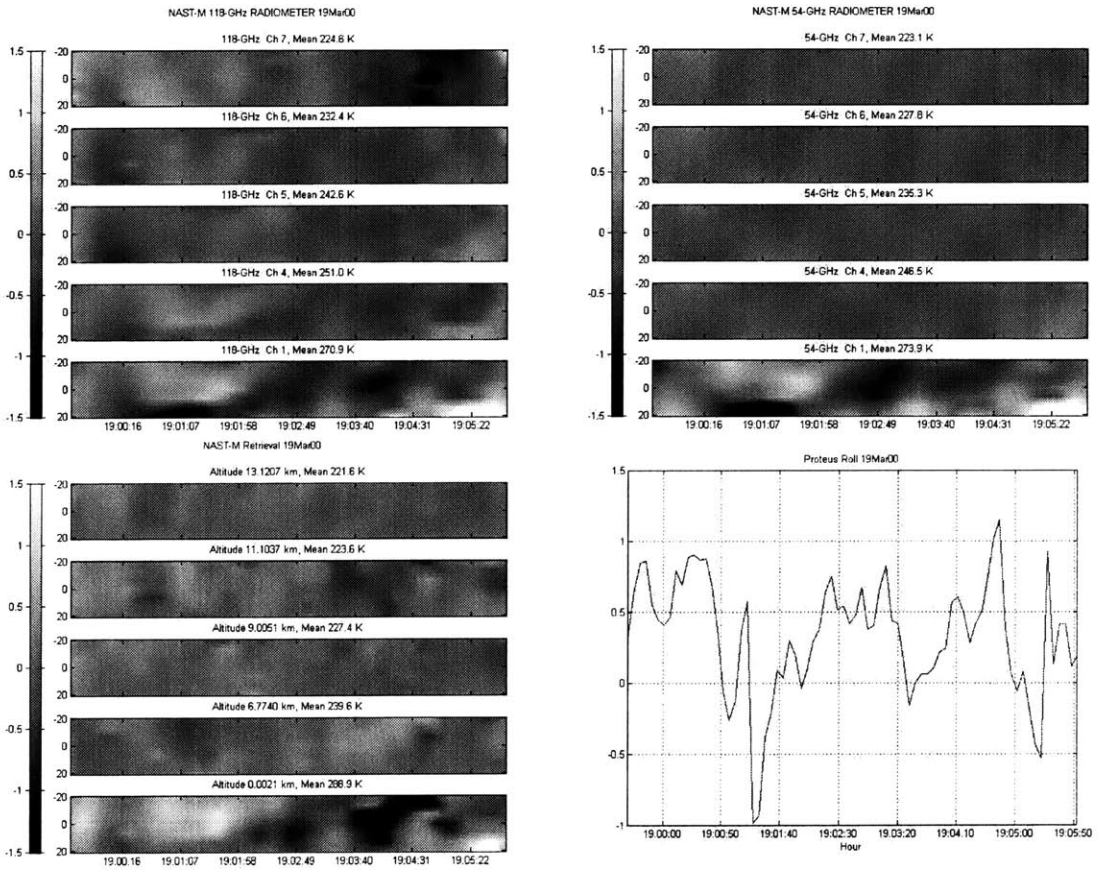


Figure C-20 19Mar00 Temperature Imagery for index 1735.

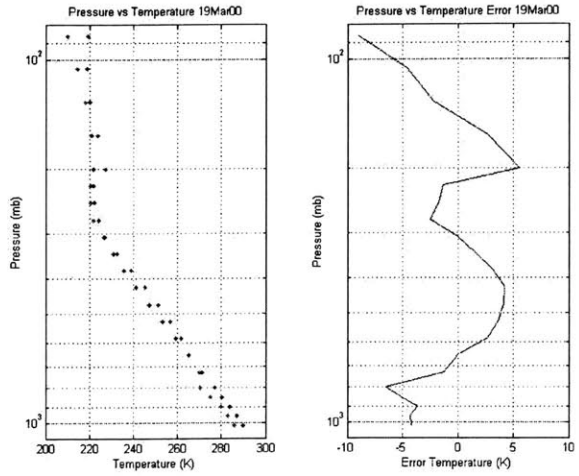


Figure C-21 19Mar00 Temperature Retrieval Comparison for index 2100.

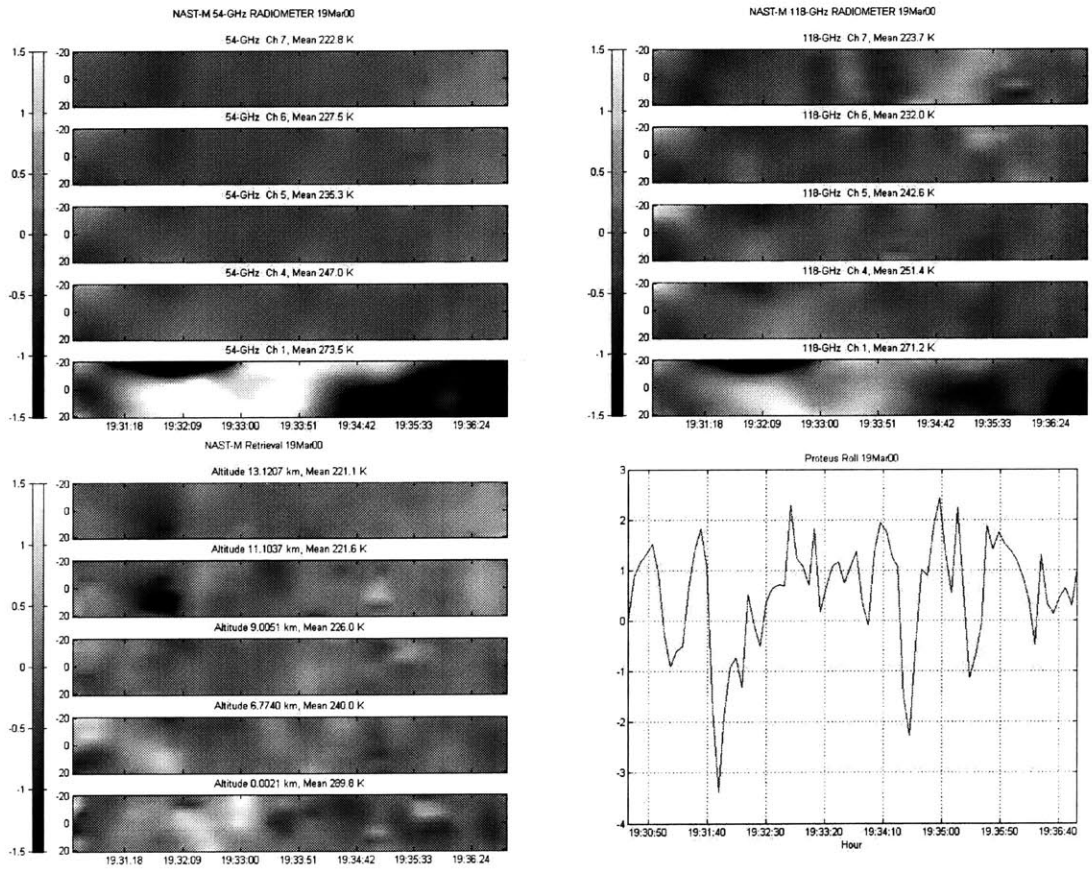


Figure C-22 19Mar00 Temperature Imagery for index 2100.

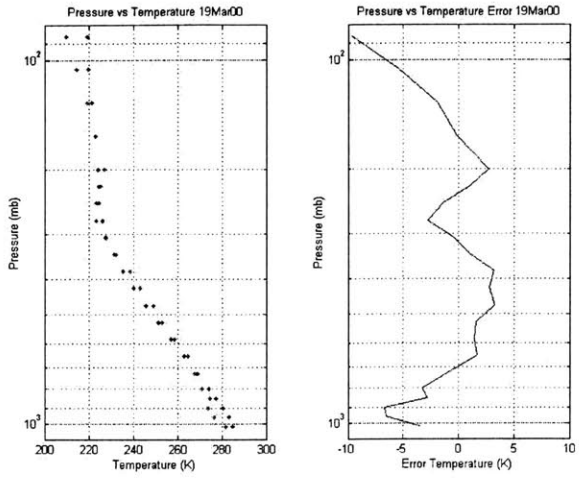


Figure C-23 19Mar00 Temperature Retrieval Comparison for index 2435.

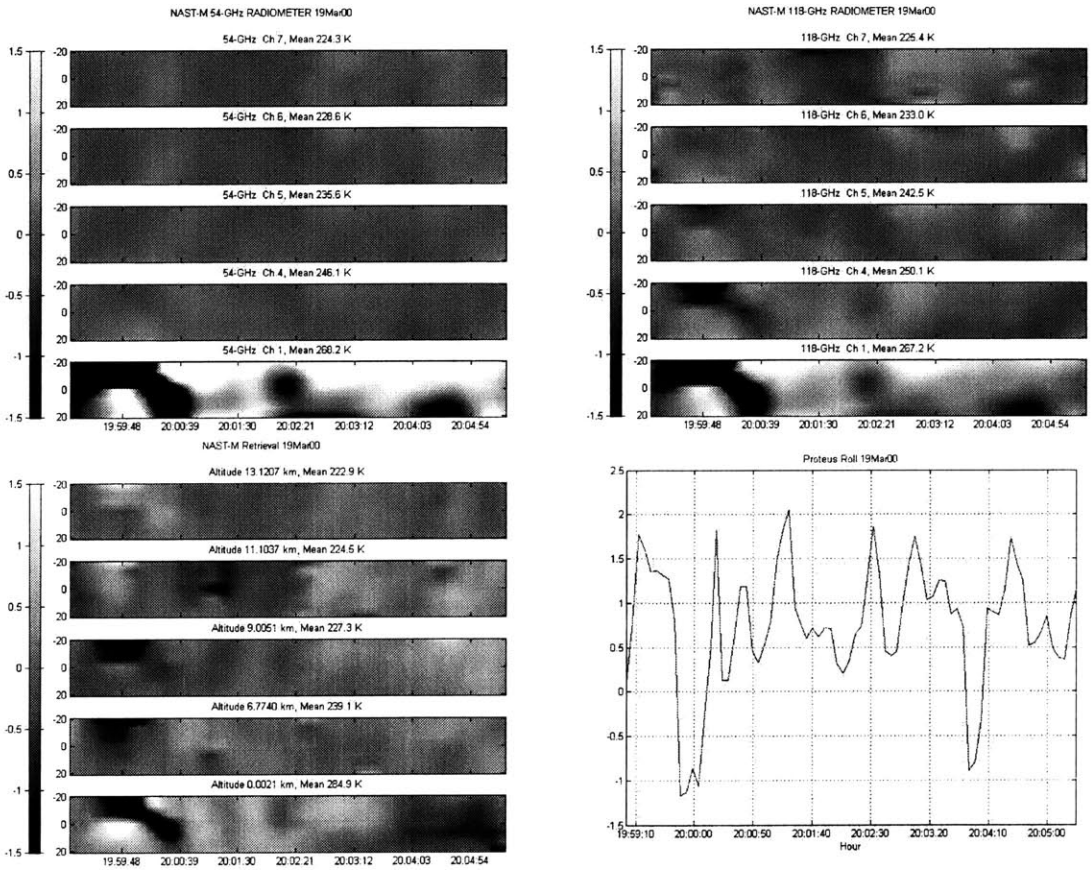


Figure C-24 19Mar00 Temperature Imagery for index 2435.

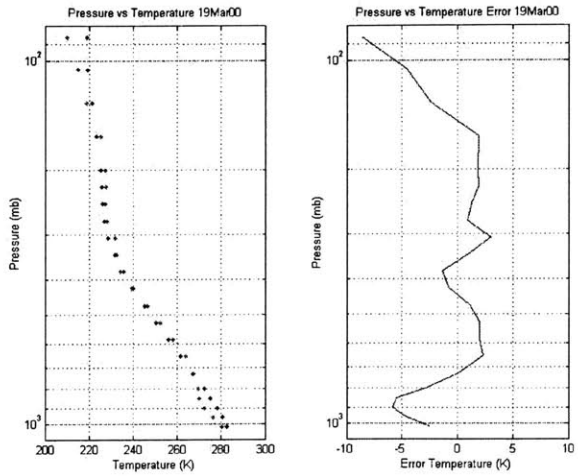


Figure C-25 19Mar00 Temperature Retrieval Comparison for index 2550.

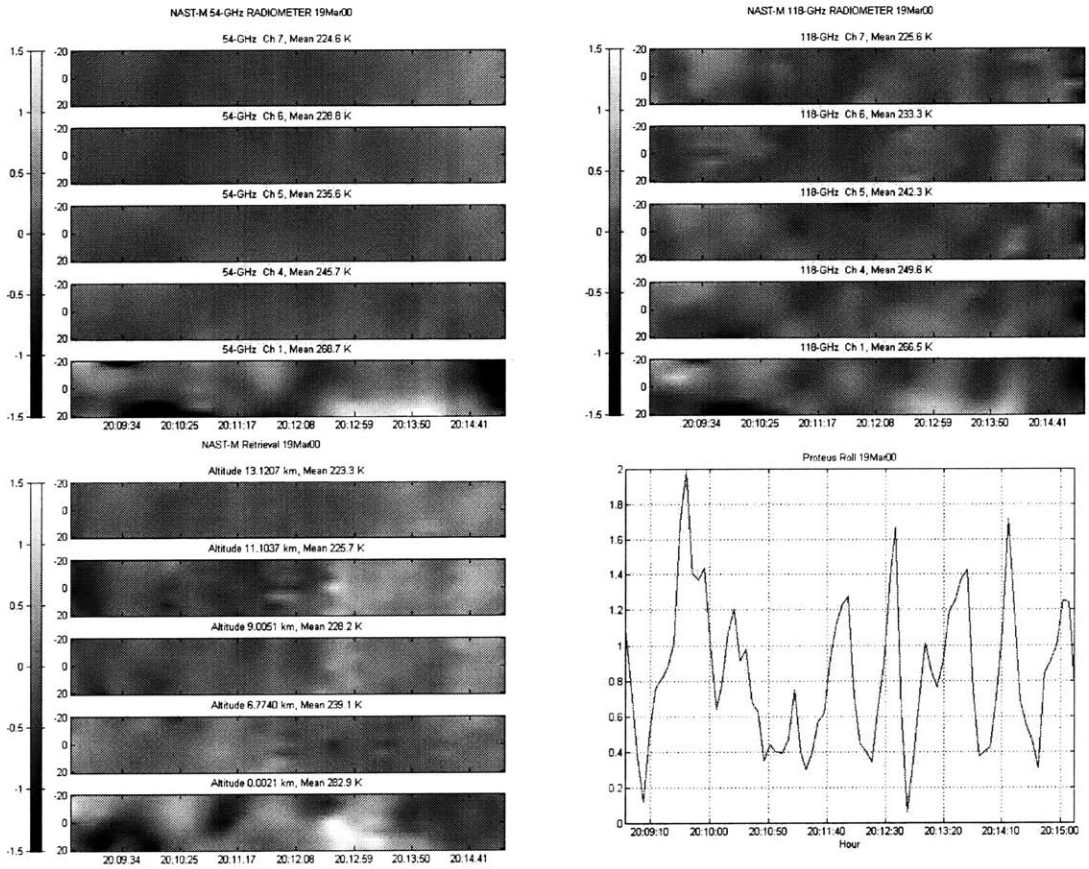


Figure C-26 19Mar00 Temperature Imagery for index 2550.

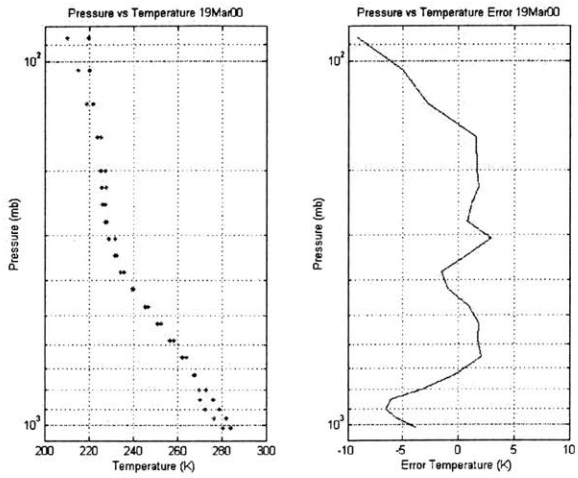


Figure C-27 19Mar00 Temperature Retrieval Comparison for index 2690.

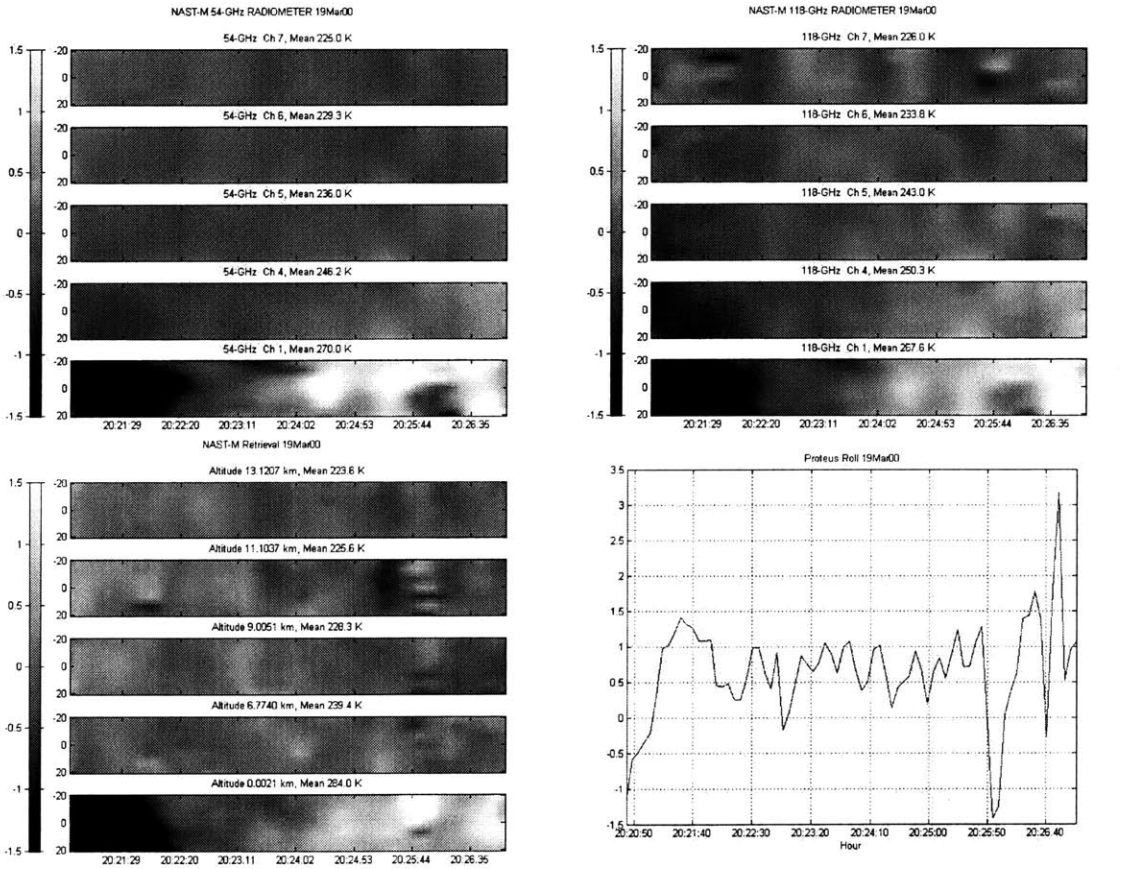


Figure C-28 19Mar00 Temperature Imagery for index 2690.

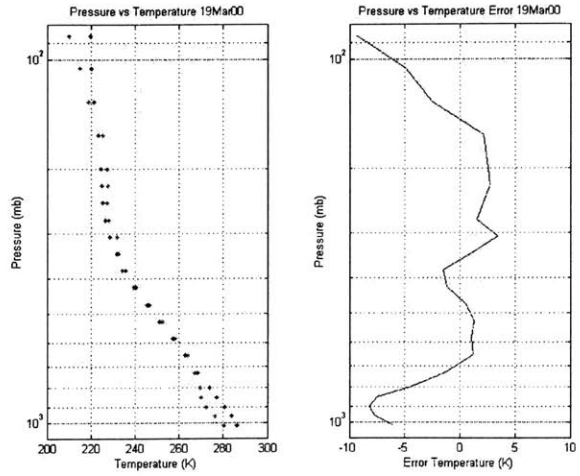


Figure C-29 19Mar00 Temperature Retrieval Comparison for index 2750.

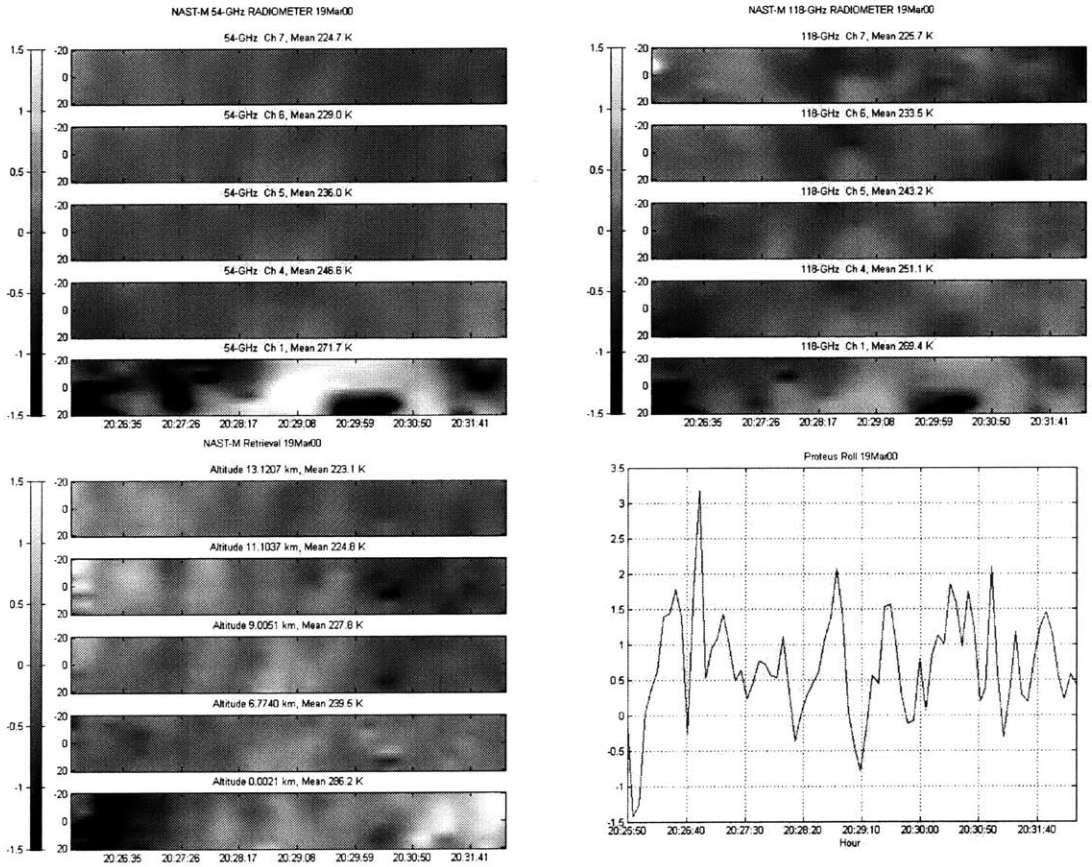


Figure C-30 19Mar00 Temperature Imagery for index 2750.

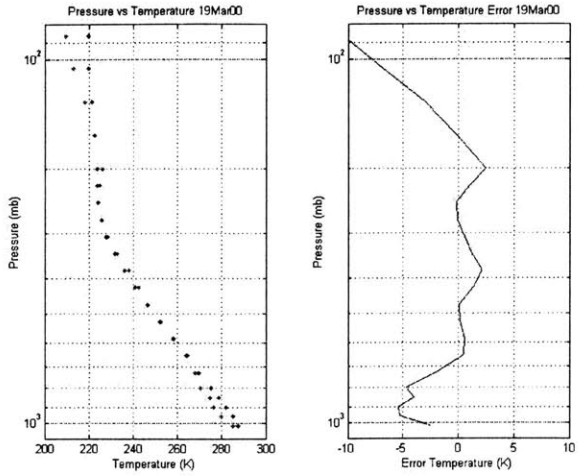


Figure C-31 19Mar00 Temperature Retrieval Comparison for index 2835.

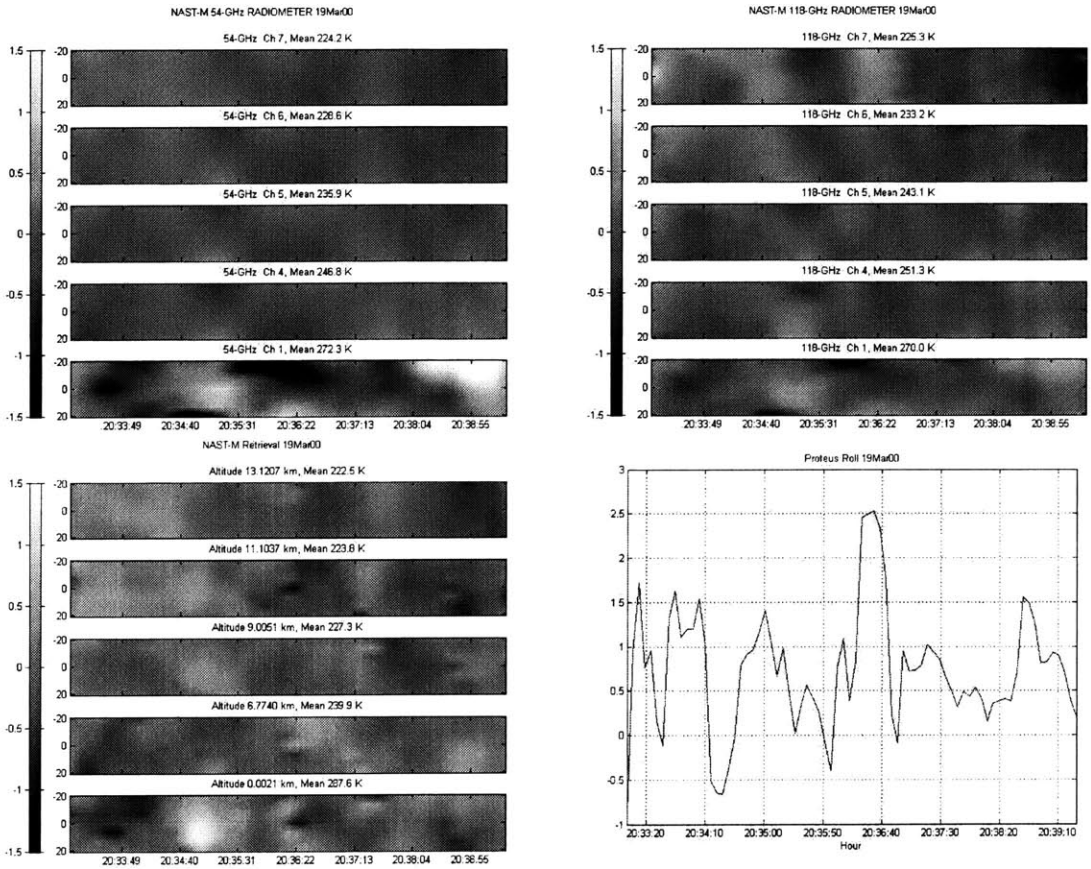


Figure C-32 19Mar00 Temperature Imagery for index 2835.

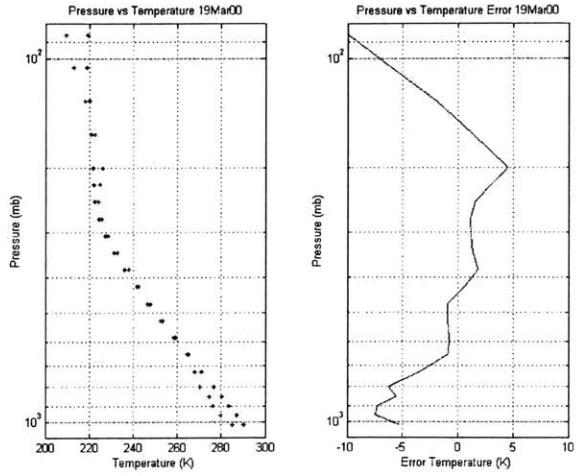


Figure C-33 19Mar00 Temperature Retrieval Comparison for index 3037.

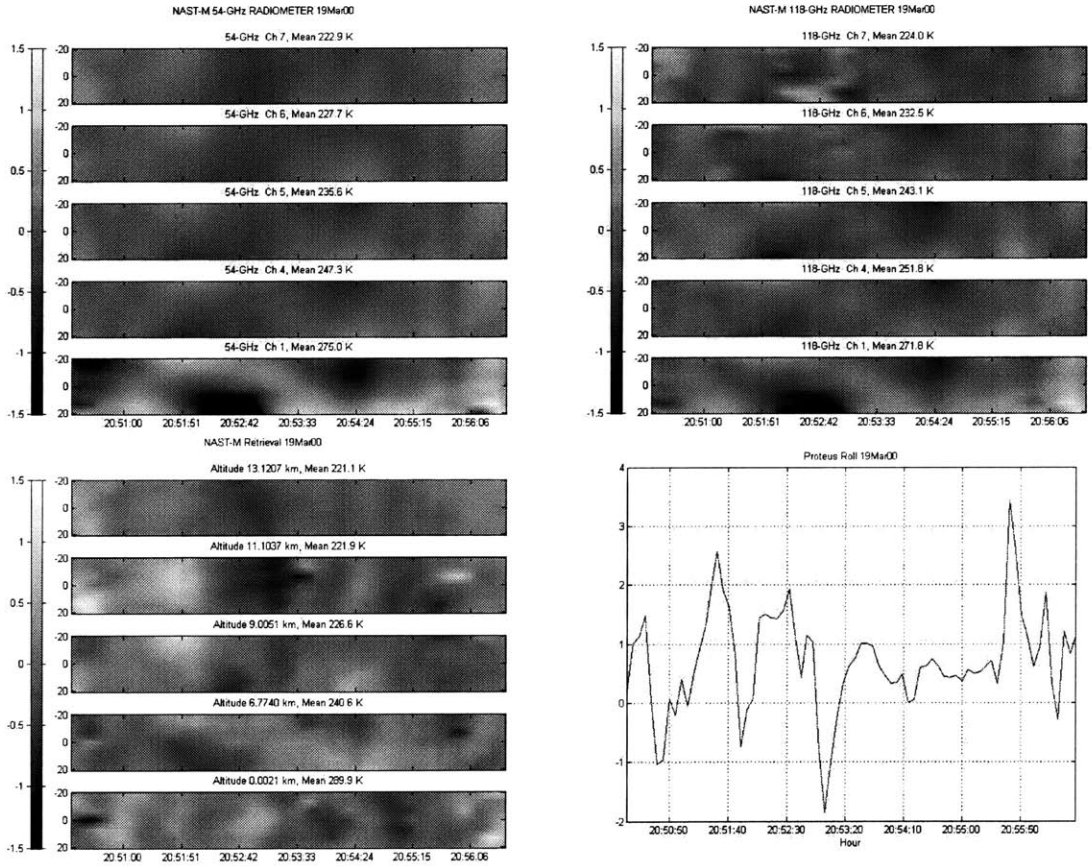


Figure C-34 19Mar00 Temperature Imagery for index 3037.

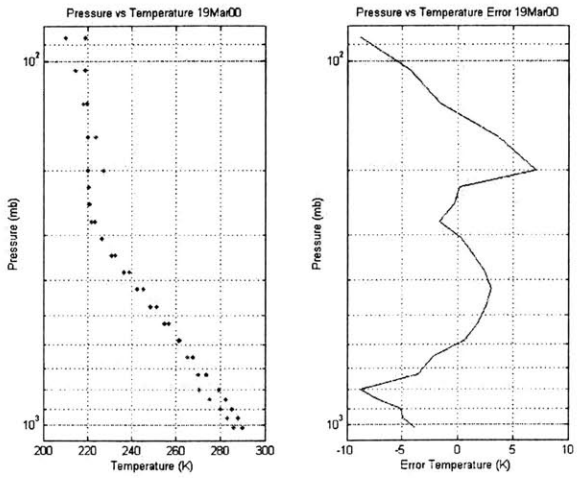


Figure C-35 19Mar00 Temperature Retrieval Comparison for index 3320.

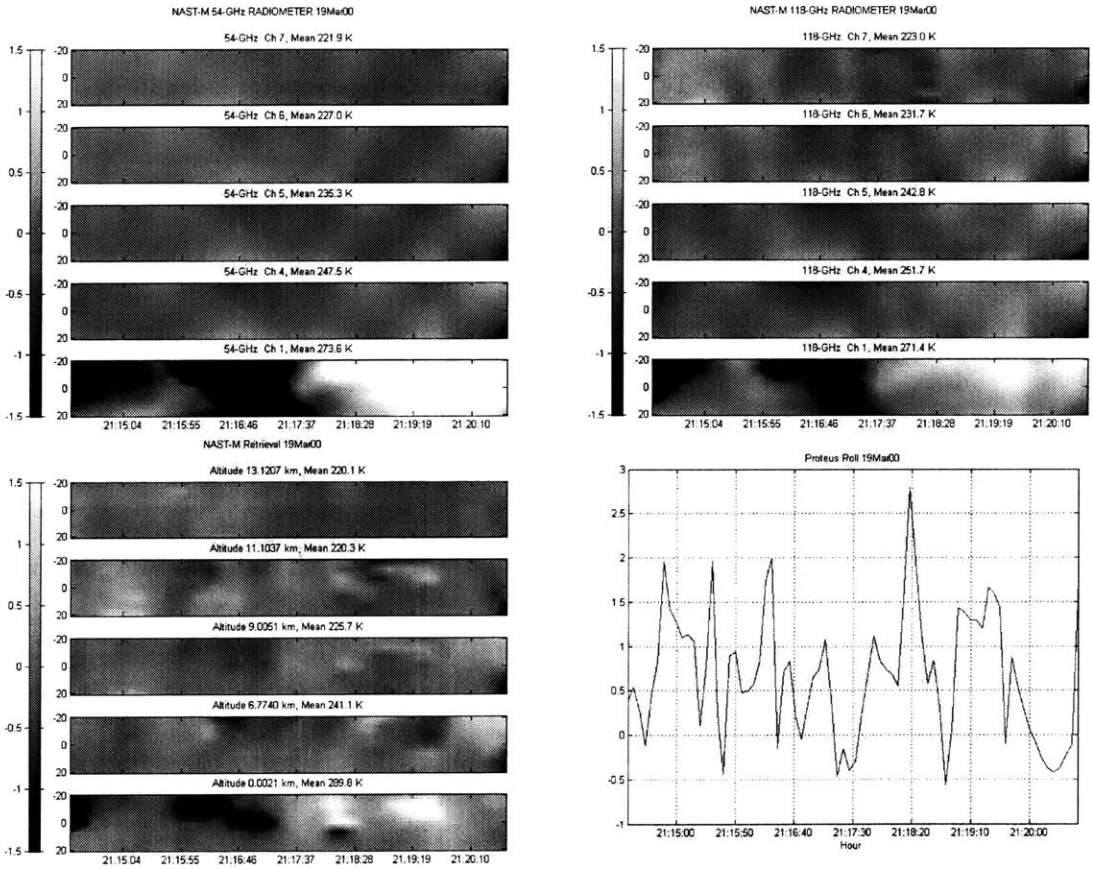


Figure C-36 19Mar00 Temperature Imagery for index 3320.

C.2 Additional Imagery from WVIOP

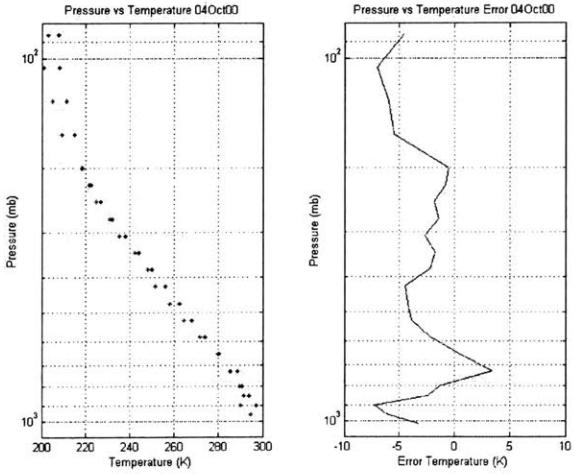


Figure C-37 04Oct00 Temperature Retrieval Comparison for index 1060.

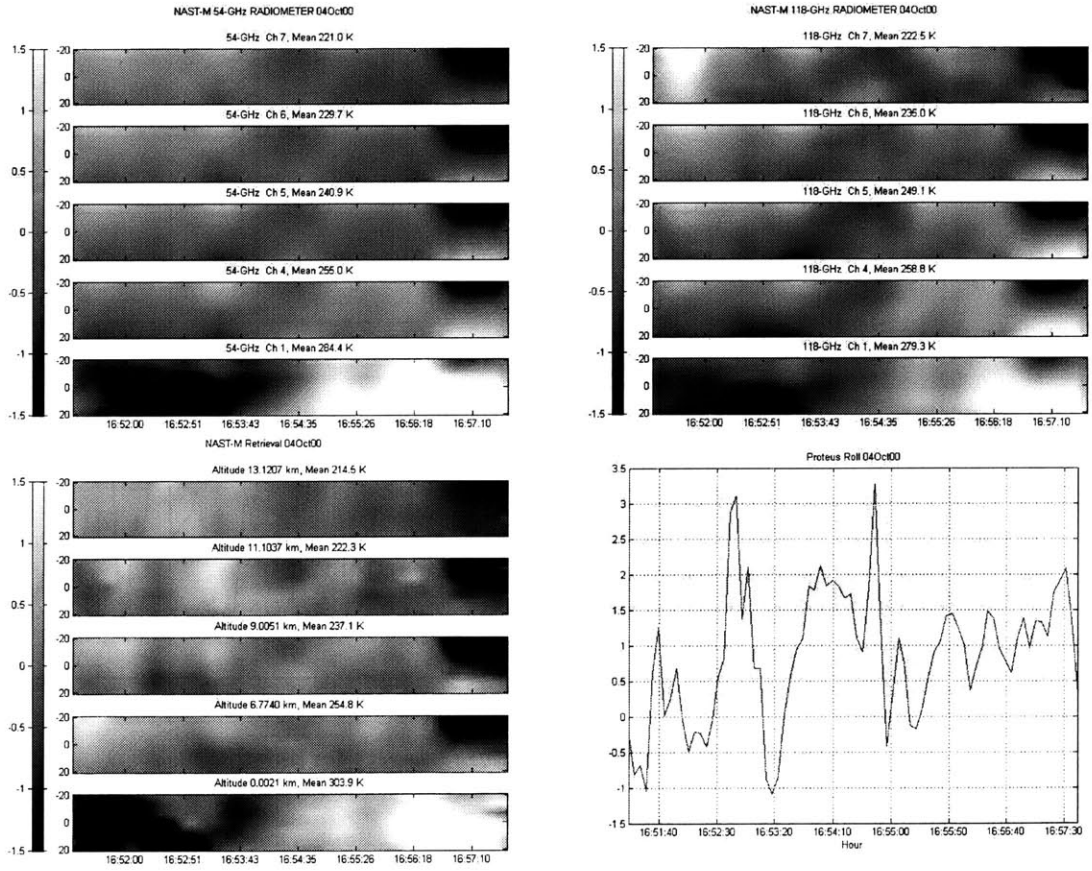


Figure C-38 04Oct00 Temperature Imagery for index 1060.

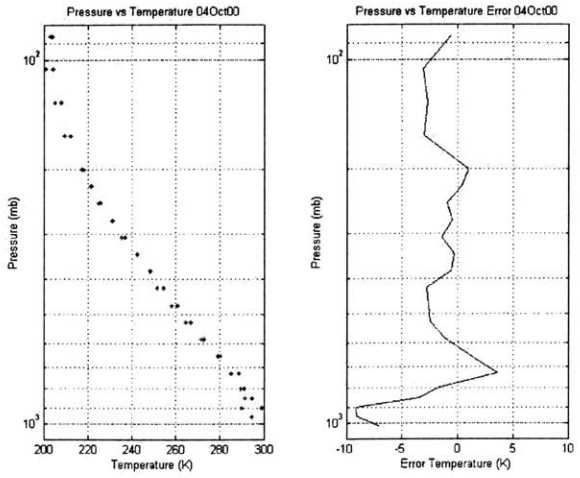


Figure C-39 04Oct00 Temperature Retrieval Comparison for index 1753.

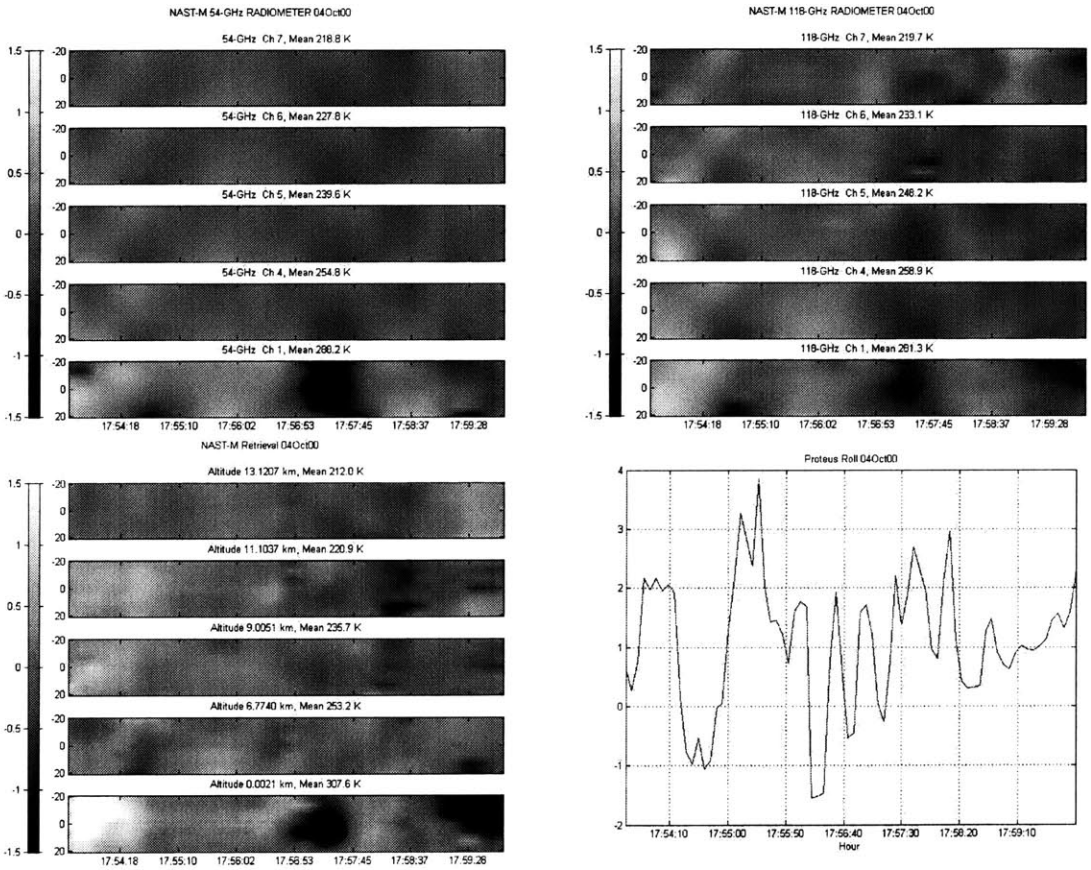


Figure C-40 04Oct00 Temperature Imagery for index 1753.

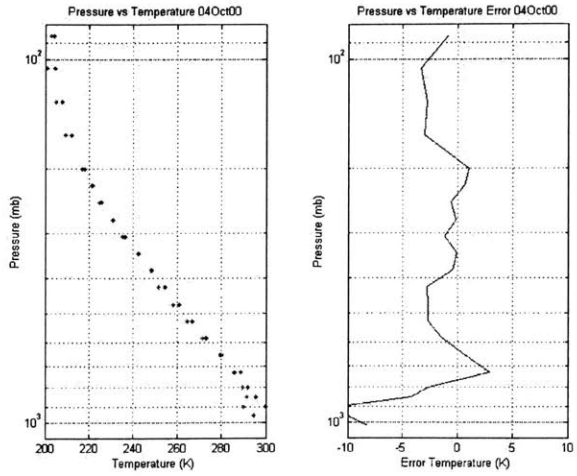


Figure C-41 04Oct00 Temperature Retrieval Comparison for index 1910.

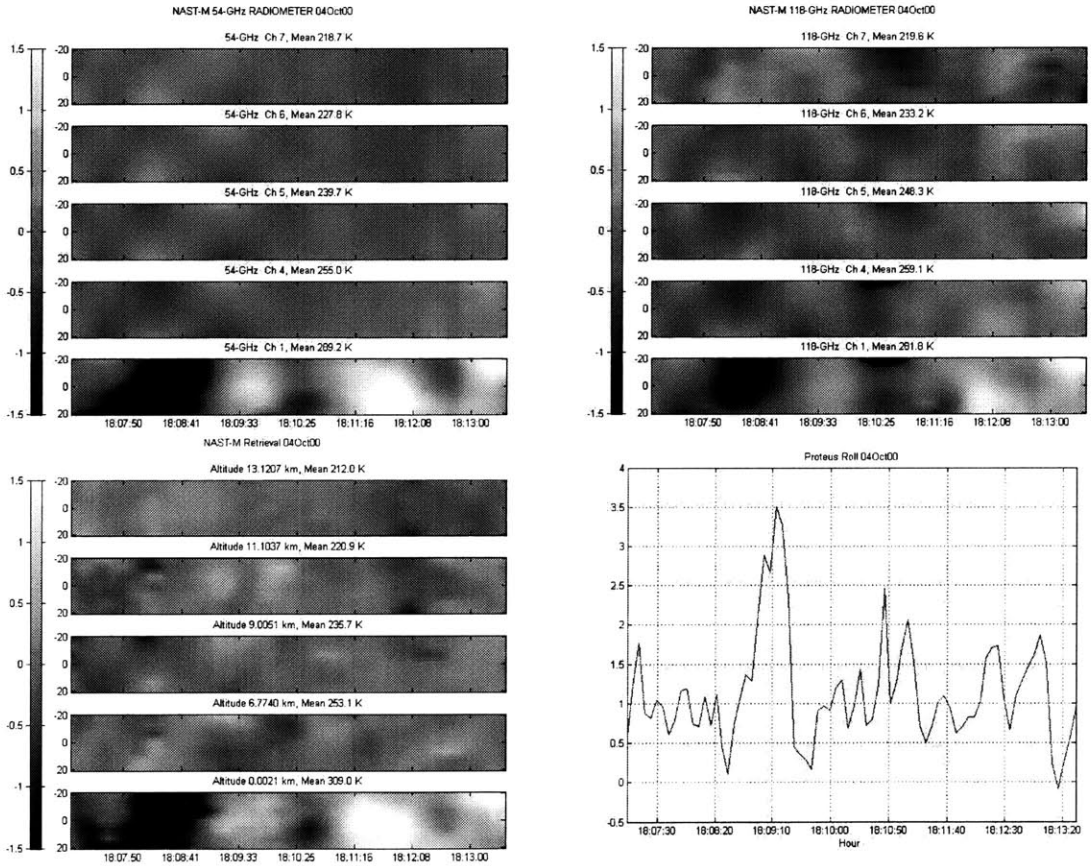


Figure C-42 04Oct00 Temperature Imagery for index 1910.

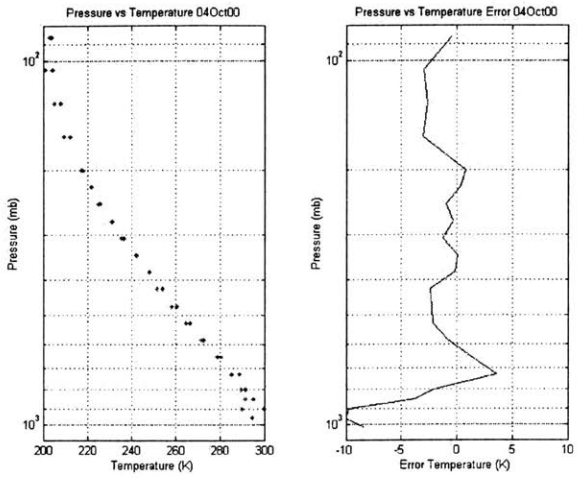


Figure C-43 04Oct00 Temperature Retrieval Comparison for index 2065.

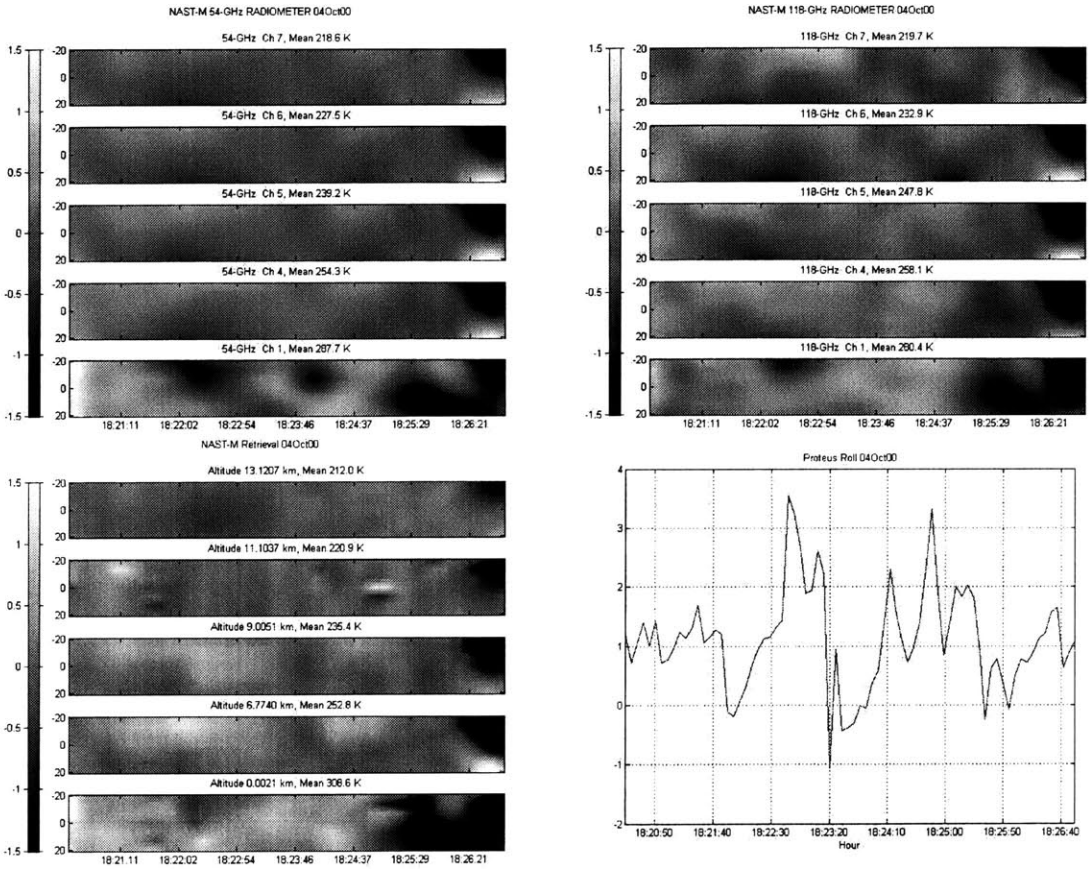


Figure C-44 04Oct00 Temperature Imagery for index 2065.

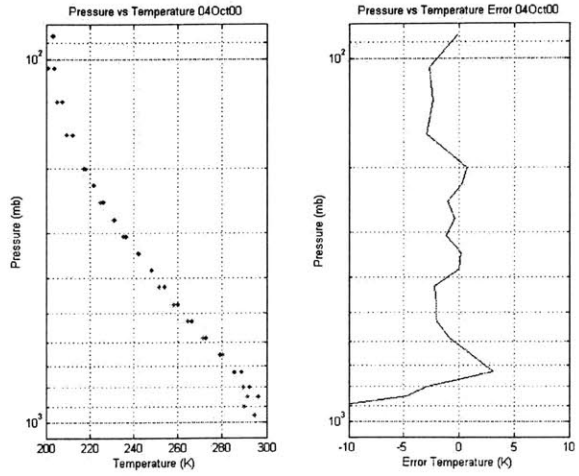


Figure C-45 04Oct00 Temperature Retrieval Comparison for index 2220.

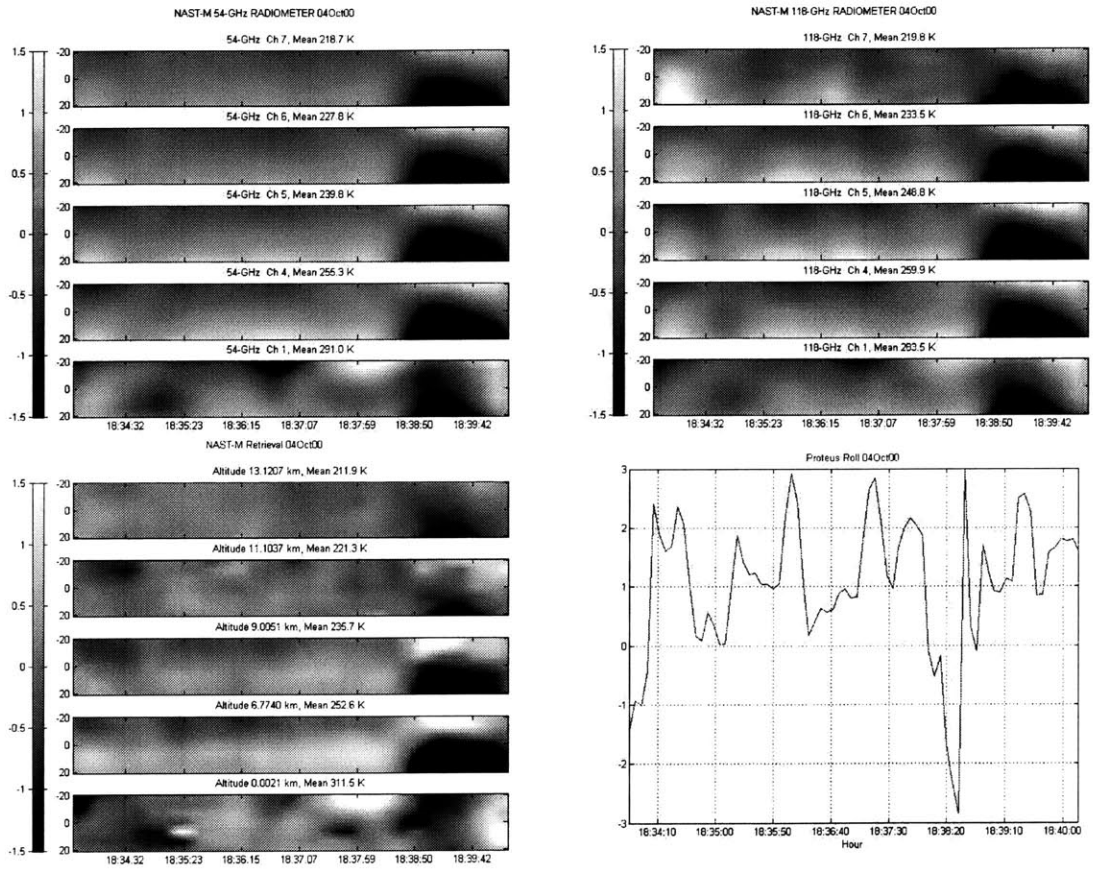


Figure C-46 04Oct00 Temperature Imagery for index 2220.

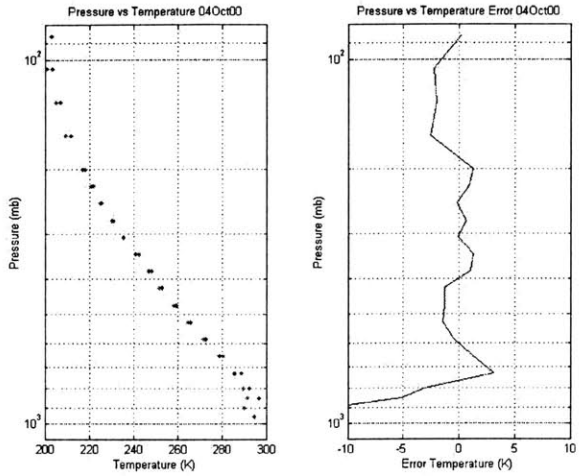


Figure C-47 04Oct00 Temperature Retrieval Comparison for index 2653.

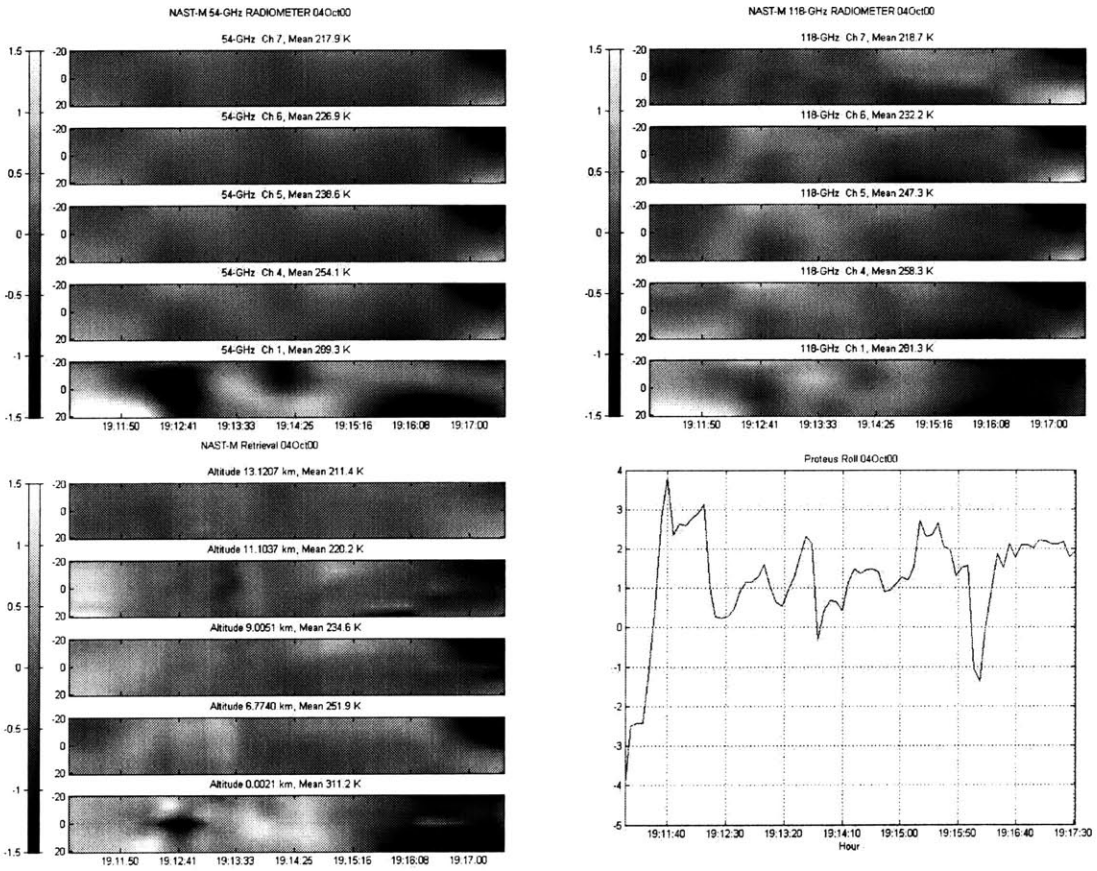


Figure C-48 04Oct00 Temperature Imagery for index 2653.

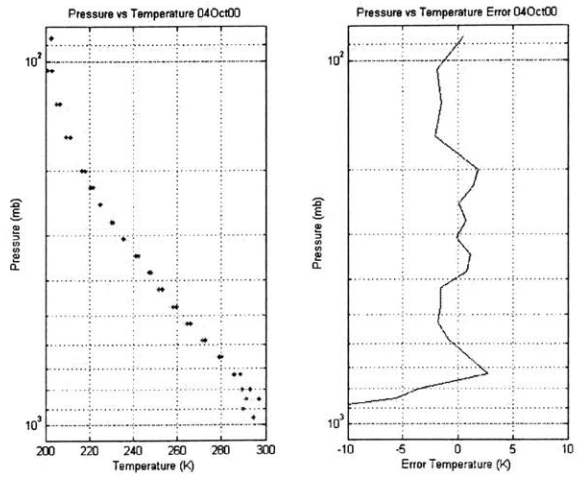


Figure C-49 04Oct00 Temperature Retrieval Comparison for index 2930.

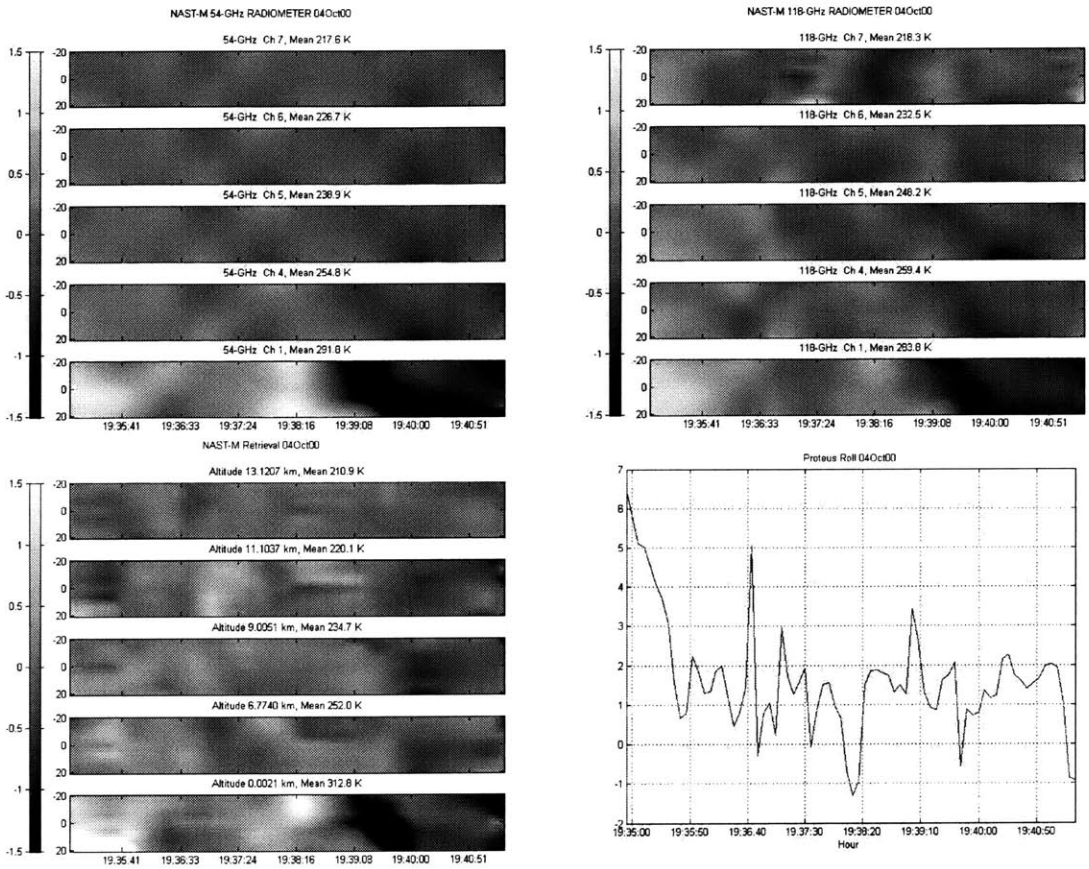


Figure C-50 04Oct00 Temperature Imagery for index 2930.

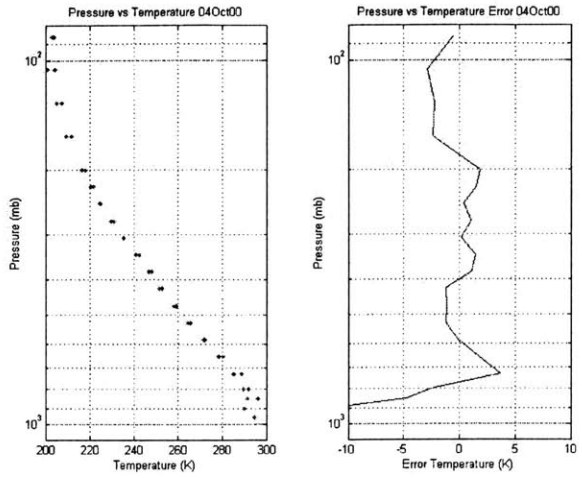


Figure C-51 04Oct00 Temperature Retrieval Comparison for index 3275.

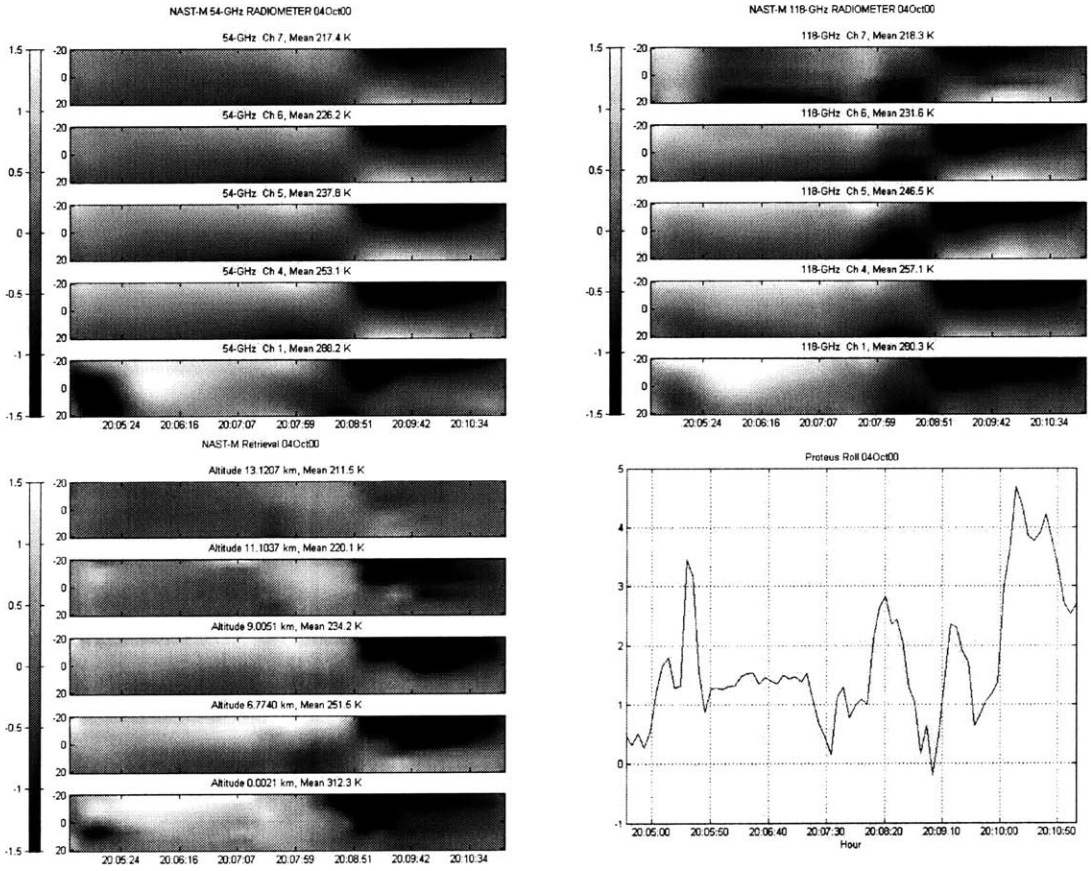


Figure C-52 04Oct00 Temperature Imagery for index 3275.

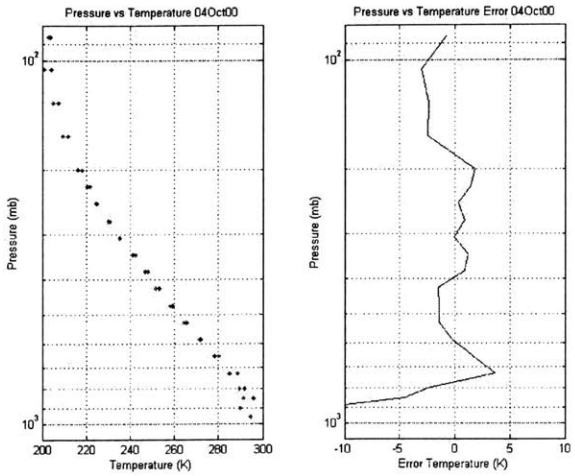


Figure C-53 04Oct00 Temperature Retrieval Comparison for index 3878.

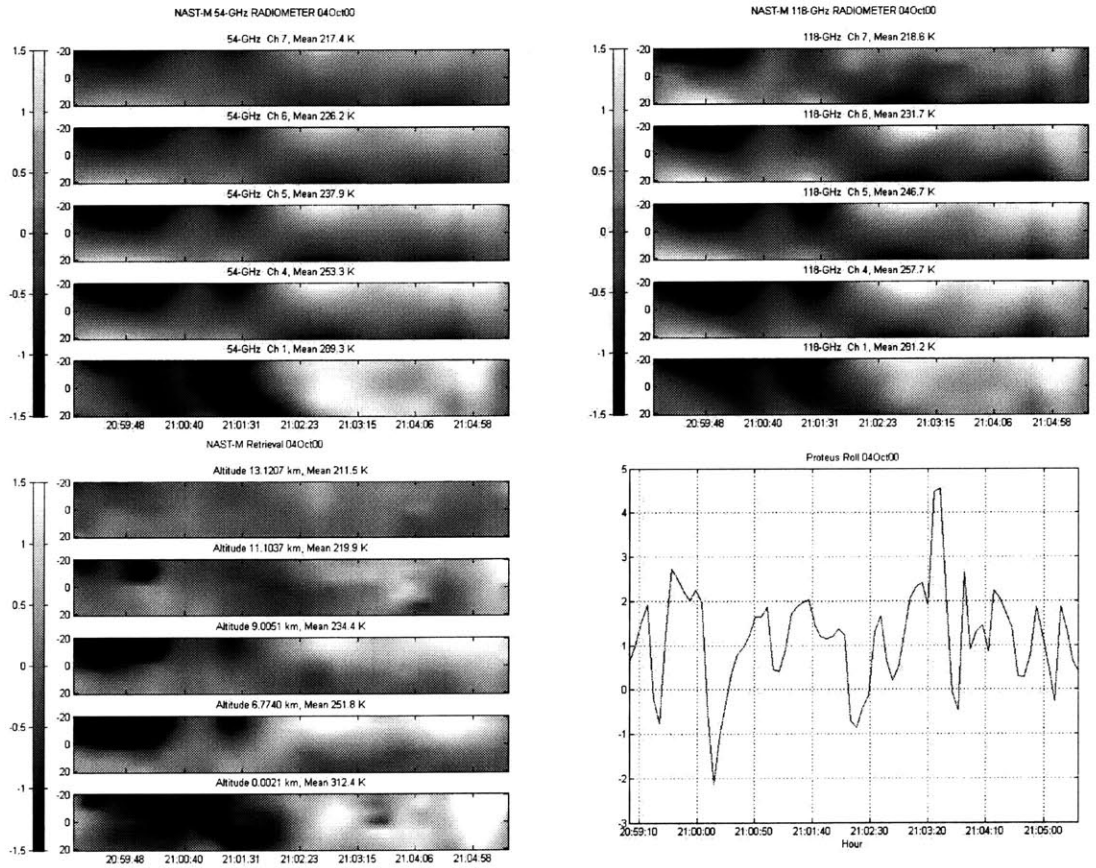


Figure C-54 04Oct00 Temperature Imagery for index 3878.

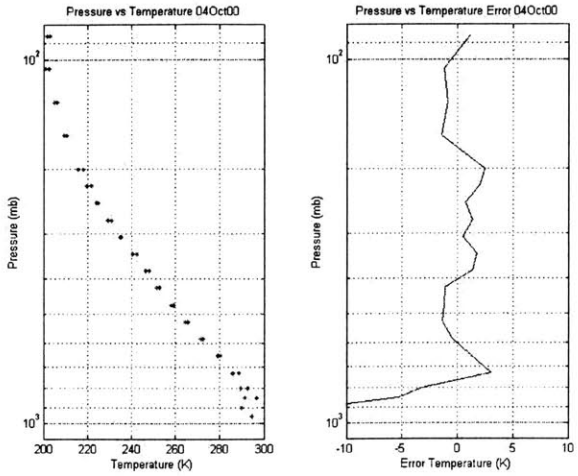


Figure C-55 04Oct00 Temperature Retrieval Comparison for index 4210.

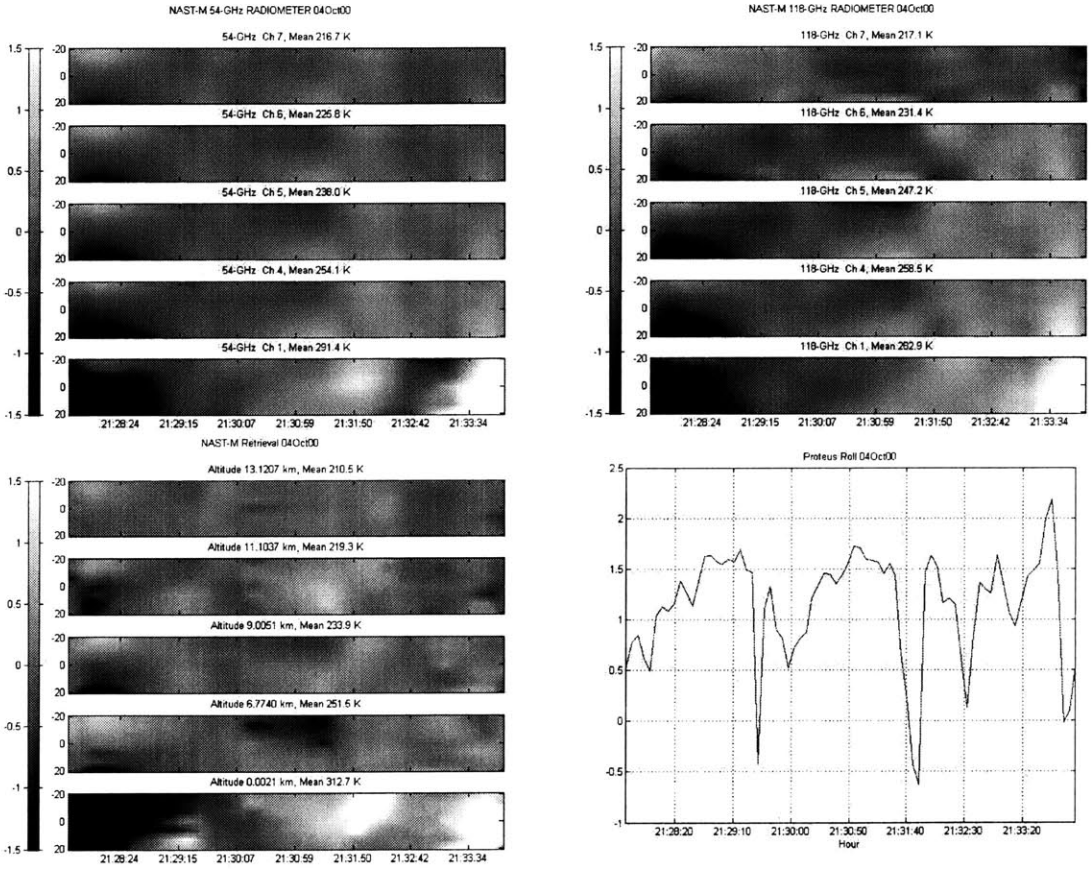


Figure C-56 04Oct00 Temperature Imagery for index 4210.

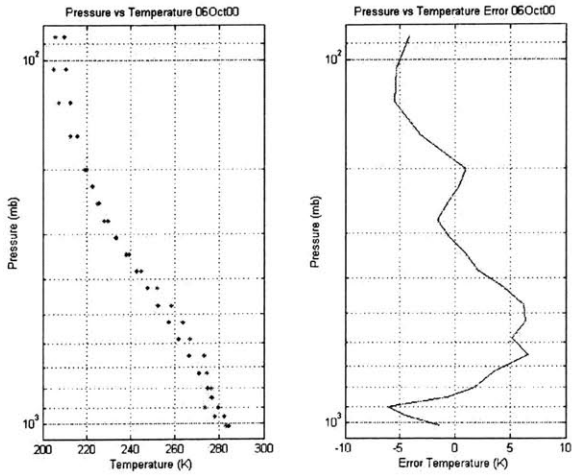


Figure C-57 06Oct00 Temperature Retrieval Comparison for index 750.

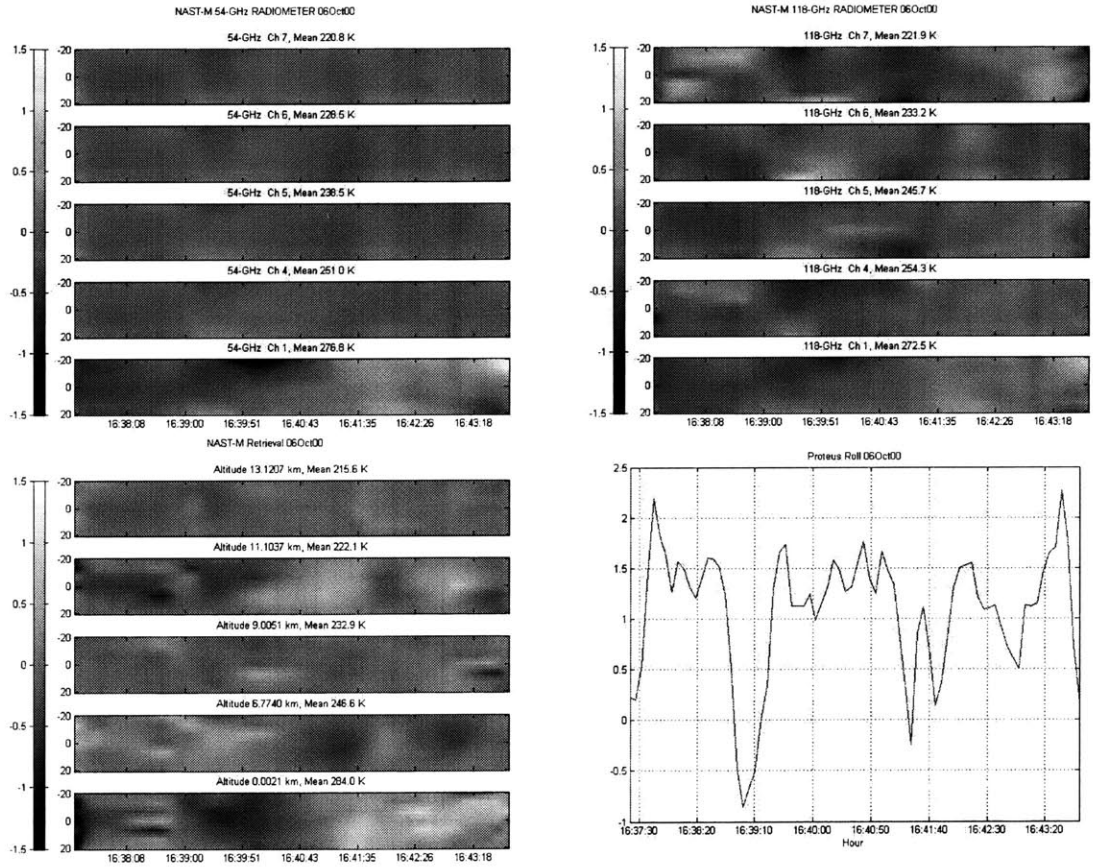


Figure C-58 06Oct00 Temperature Imagery for index 750.

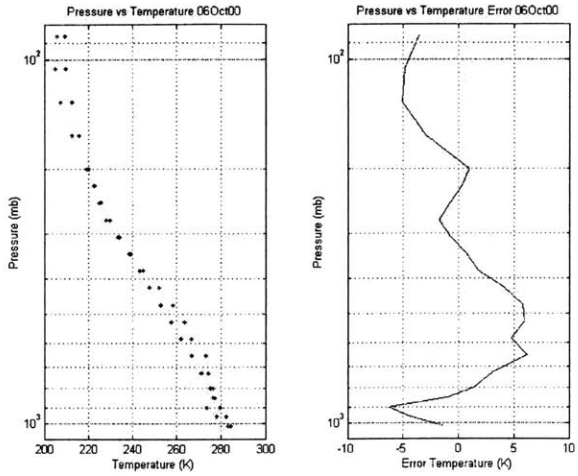


Figure C-59 06Oct00 Temperature Retrieval Comparison for index 870.

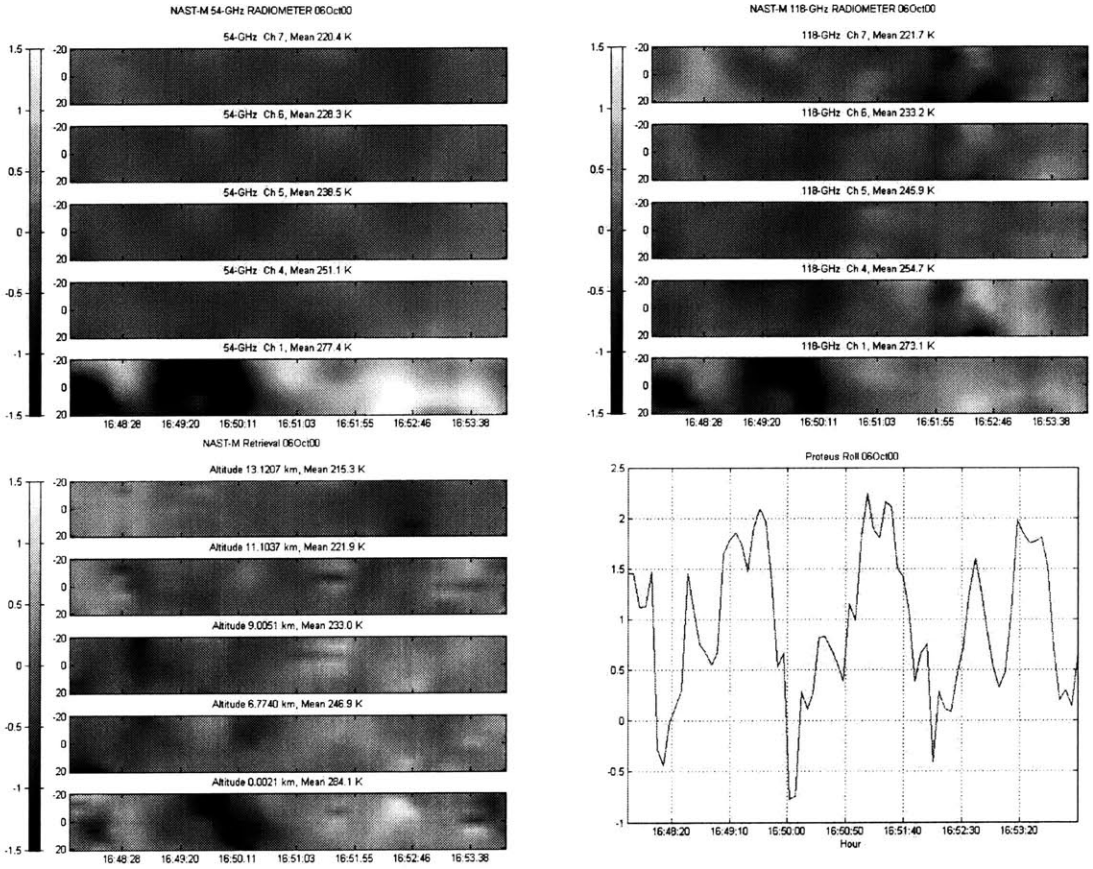


Figure C-60 06Oct00 Temperature Imagery for index 870.

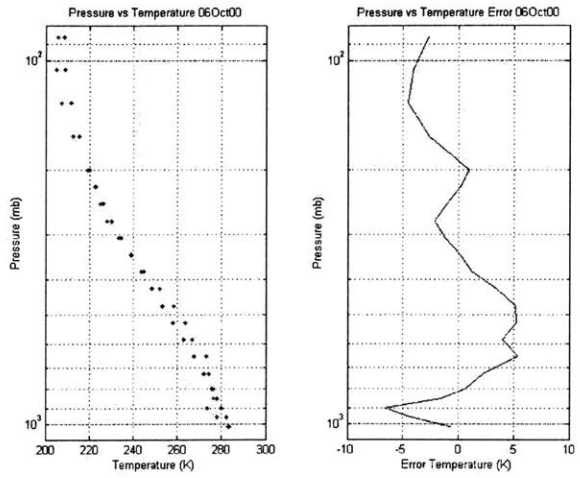


Figure C-61 06Oct00 Temperature Retrieval Comparison for index 940.

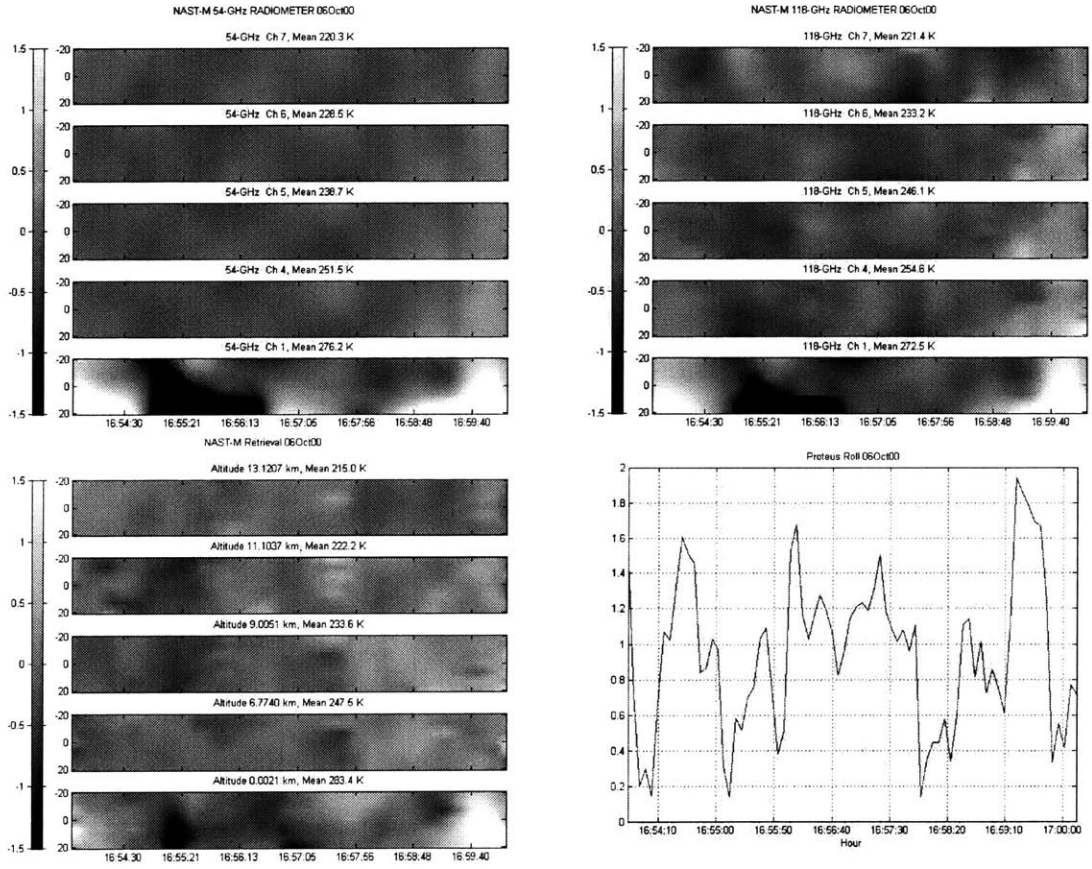


Figure C-62 06Oct00 Temperature Imagery for index 940.

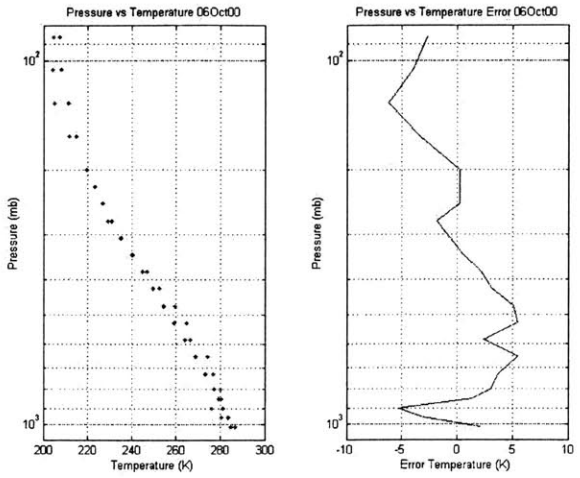


Figure C-63 06Oct00 Temperature Retrieval Comparison for index 1180.

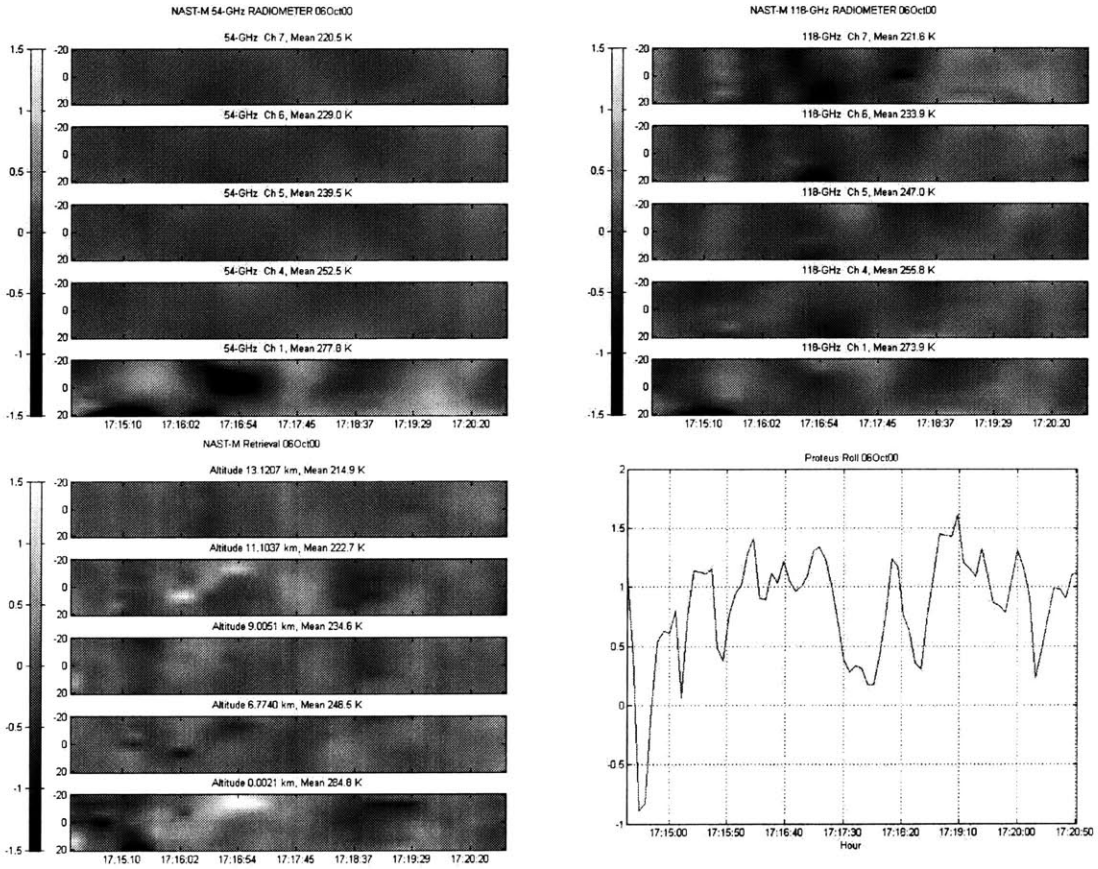


Figure C-64 06Oct00 Temperature Imagery for index 1180.

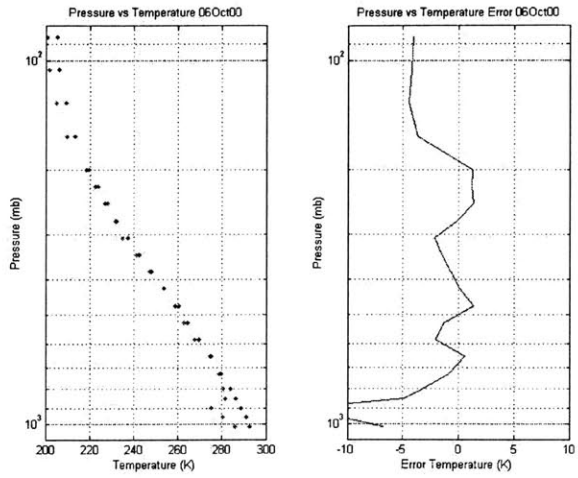


Figure C-65 06Oct00 Temperature Retrieval Comparison for index 1250.

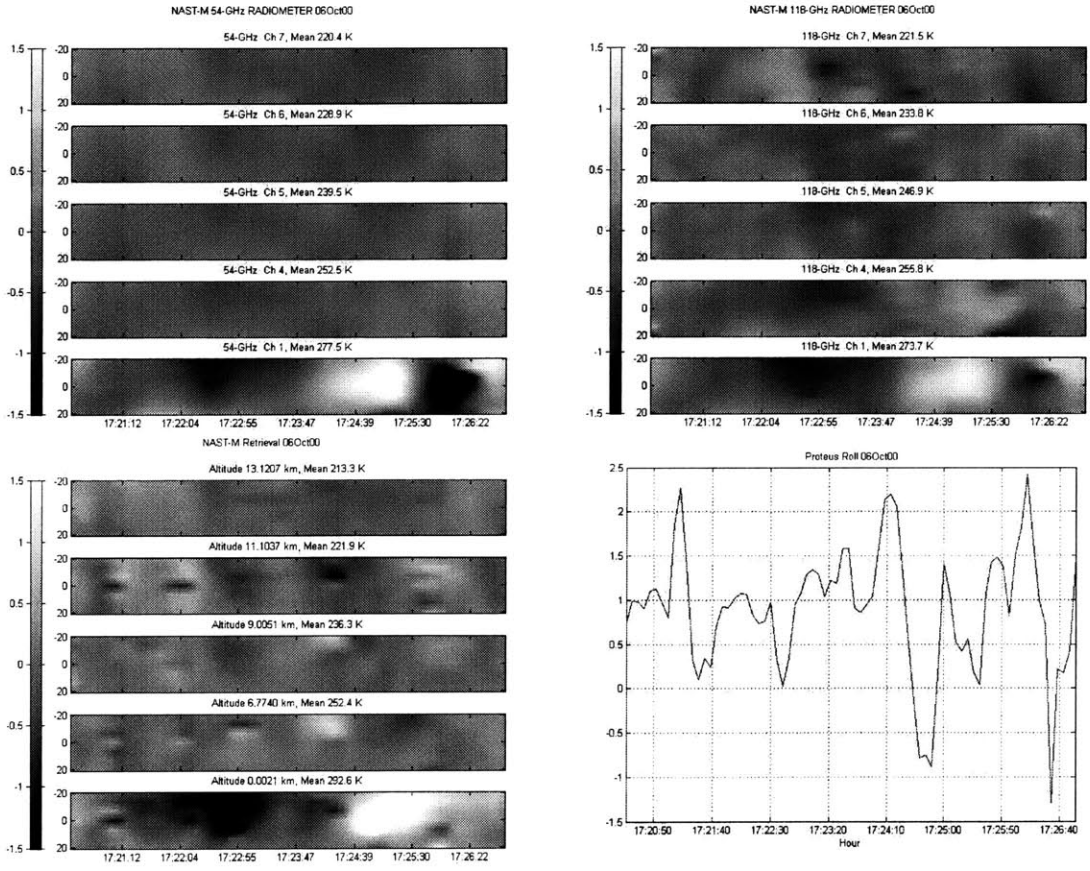


Figure C-66 06Oct00 Temperature Imagery for index 1250.

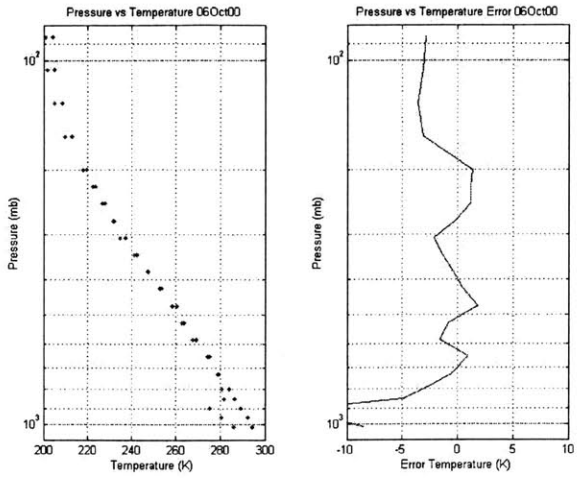


Figure C-67 06Oct00 Temperature Retrieval Comparison for index 1370.

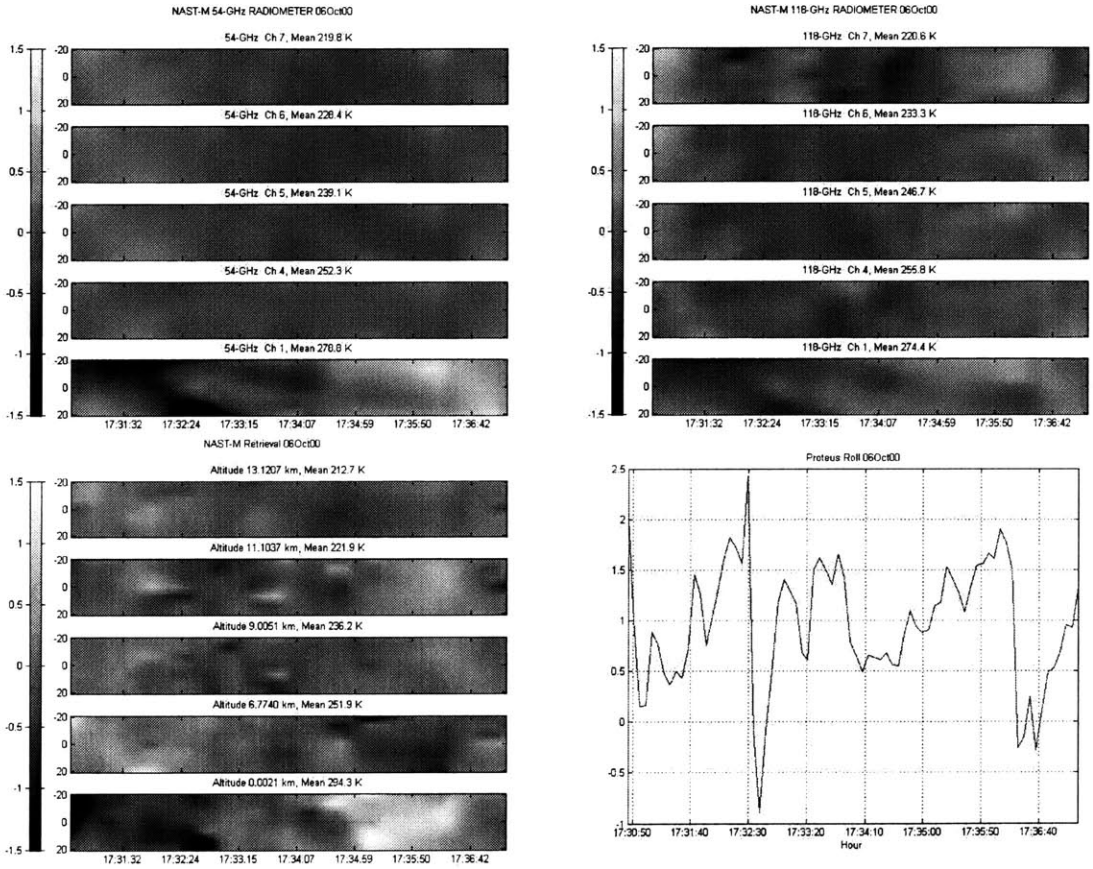


Figure C-68 06Oct00 Temperature Imagery for index 1370.

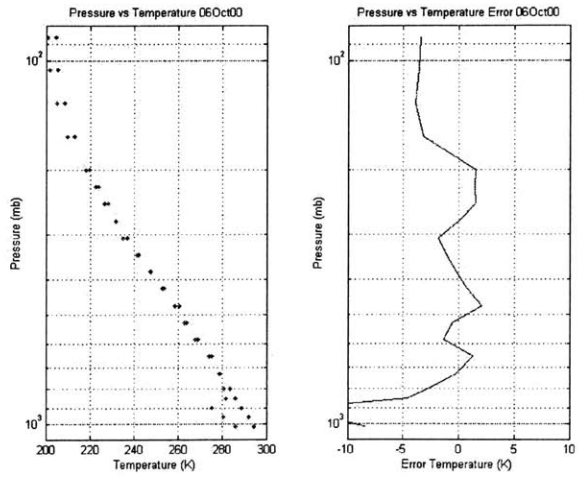


Figure C-69 06Oct00 Temperature Retrieval Comparison for index 1440.

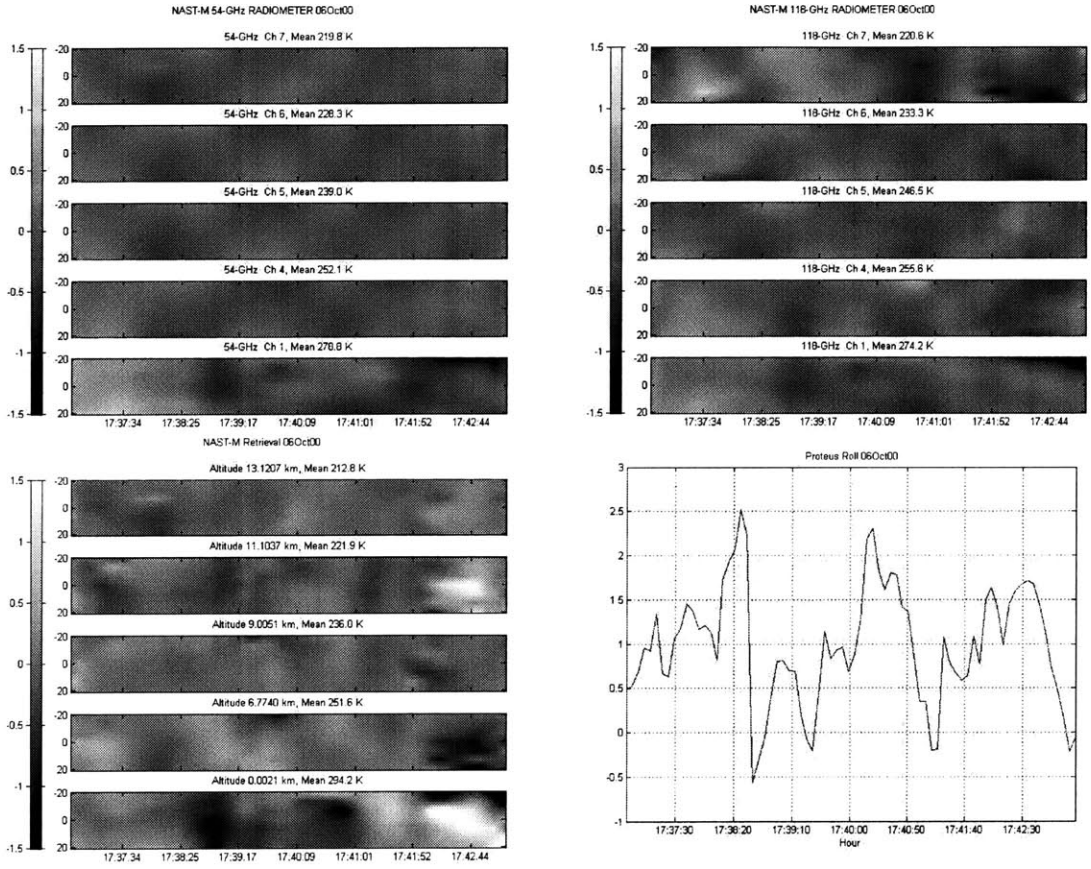


Figure C-70 06Oct00 Temperature Imagery for index 1440.

C.3 Additional Imagery from AFWEX

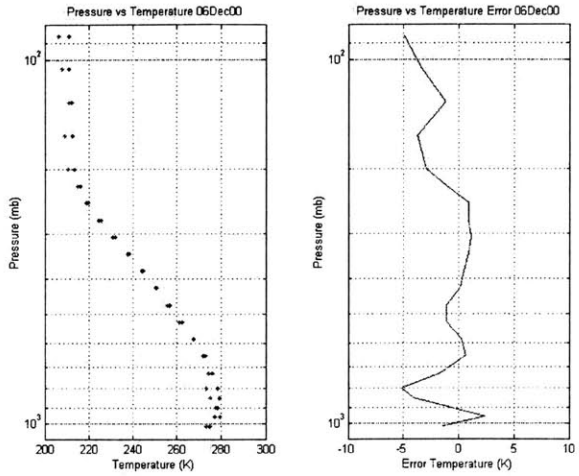


Figure C-71 06Dec00 Temperature Retrieval Comparison for index 1105.

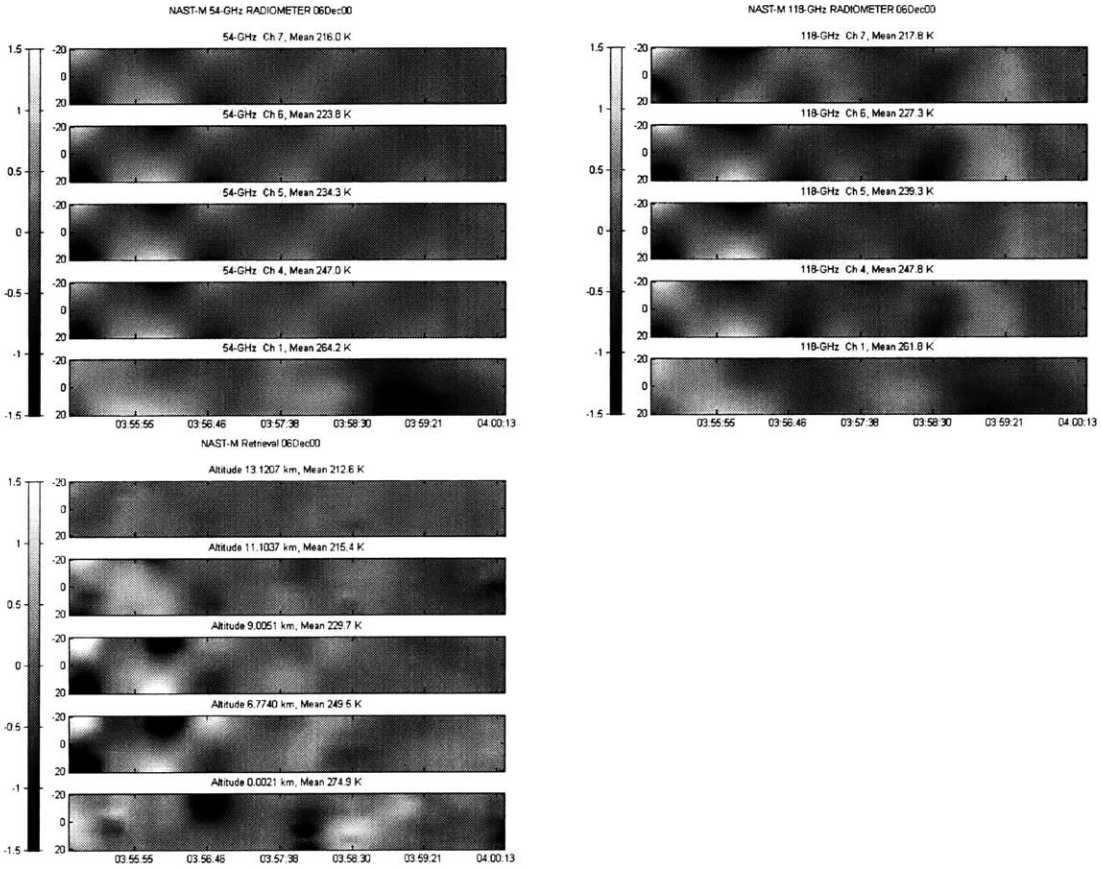


Figure C-72 06Dec00 Temperature Imagery for index 1105.

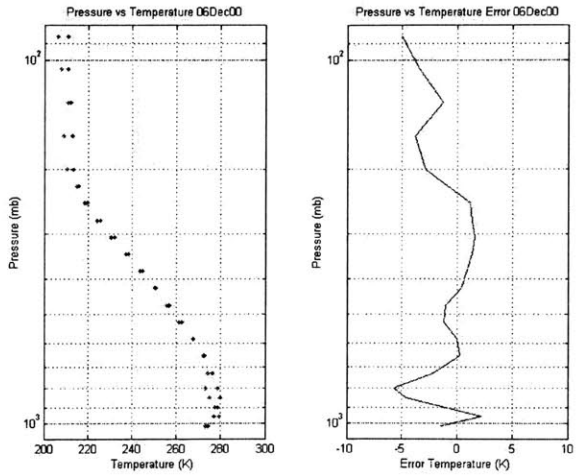


Figure C-73 06Dec00 Temperature Retrieval Comparison for index 1225.

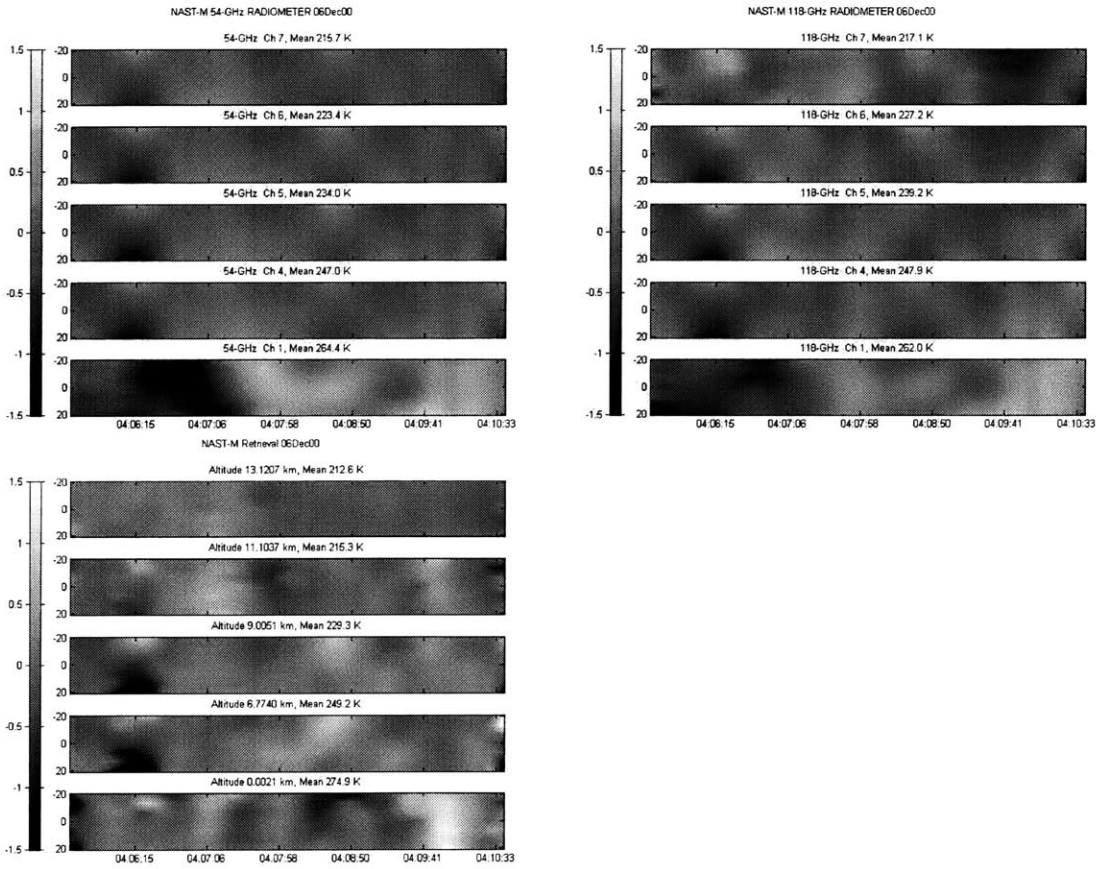


Figure C-74 06Dec00 Temperature Imagery for index 1225.

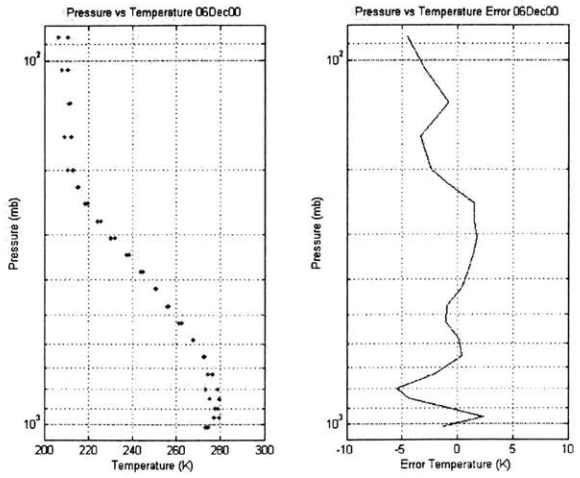


Figure C-75 06Dec00 Temperature Retrieval Comparison for index 1333.

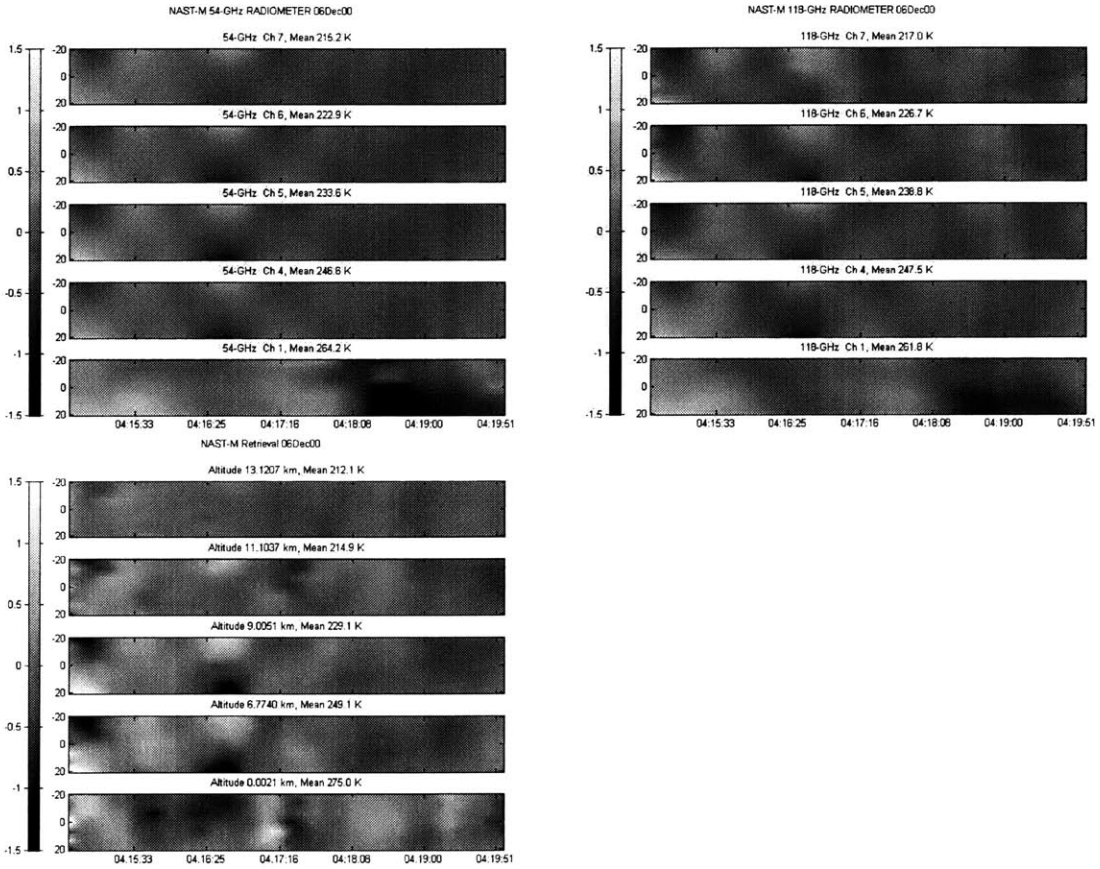


Figure C-76 06Dec00 Temperature Imagery for index 1333.

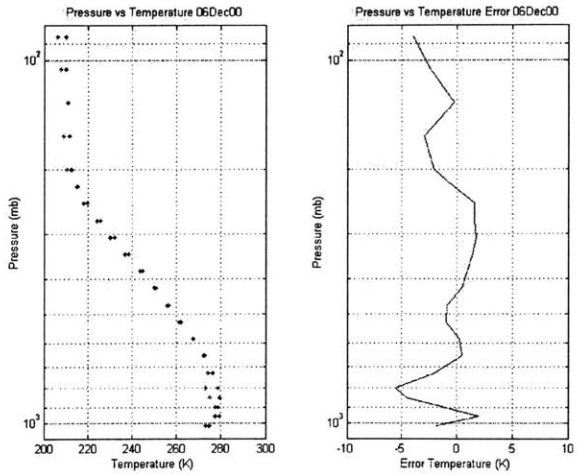


Figure C-77 06Dec00 Temperature Retrieval Comparison for index 1453.

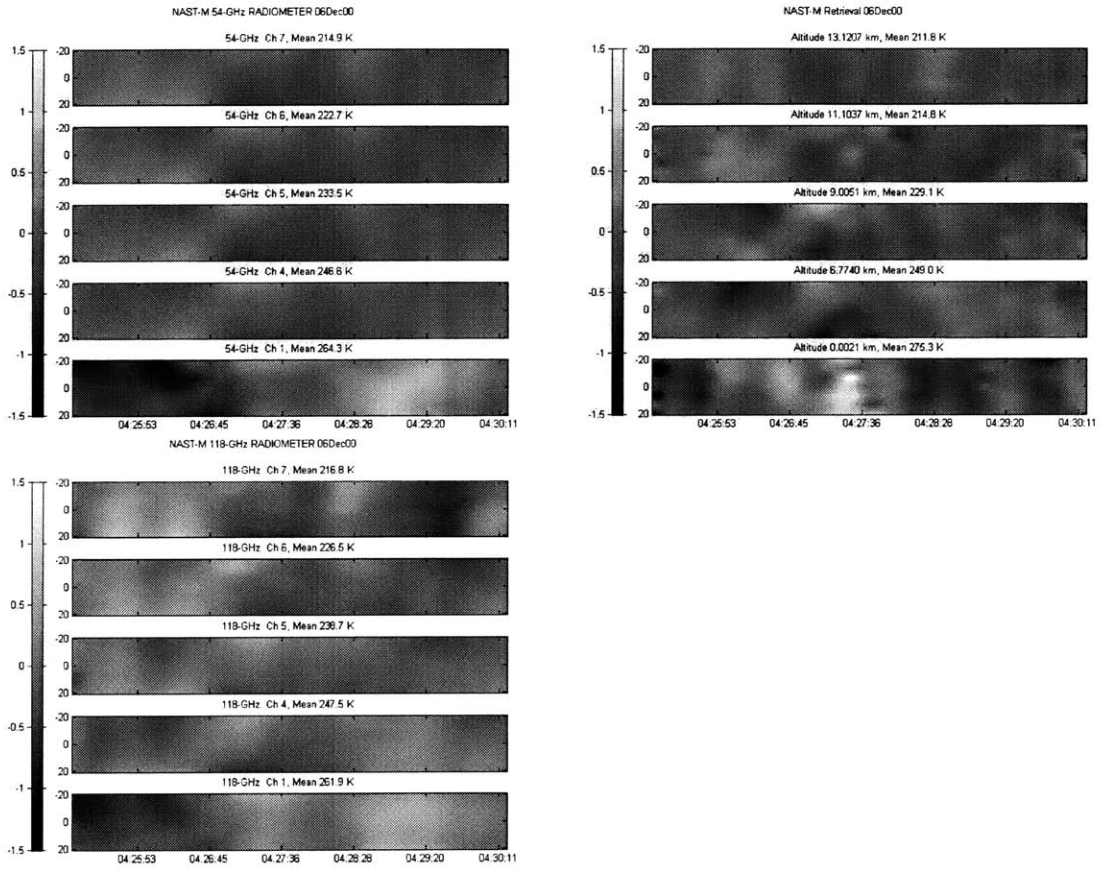


Figure C-78 06Dec00 Temperature Imagery for index 1453.

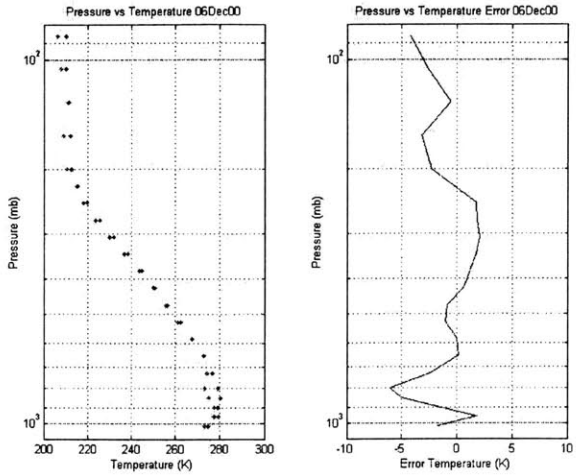


Figure C-79 06Dec00 Temperature Retrieval Comparison for index 1669.

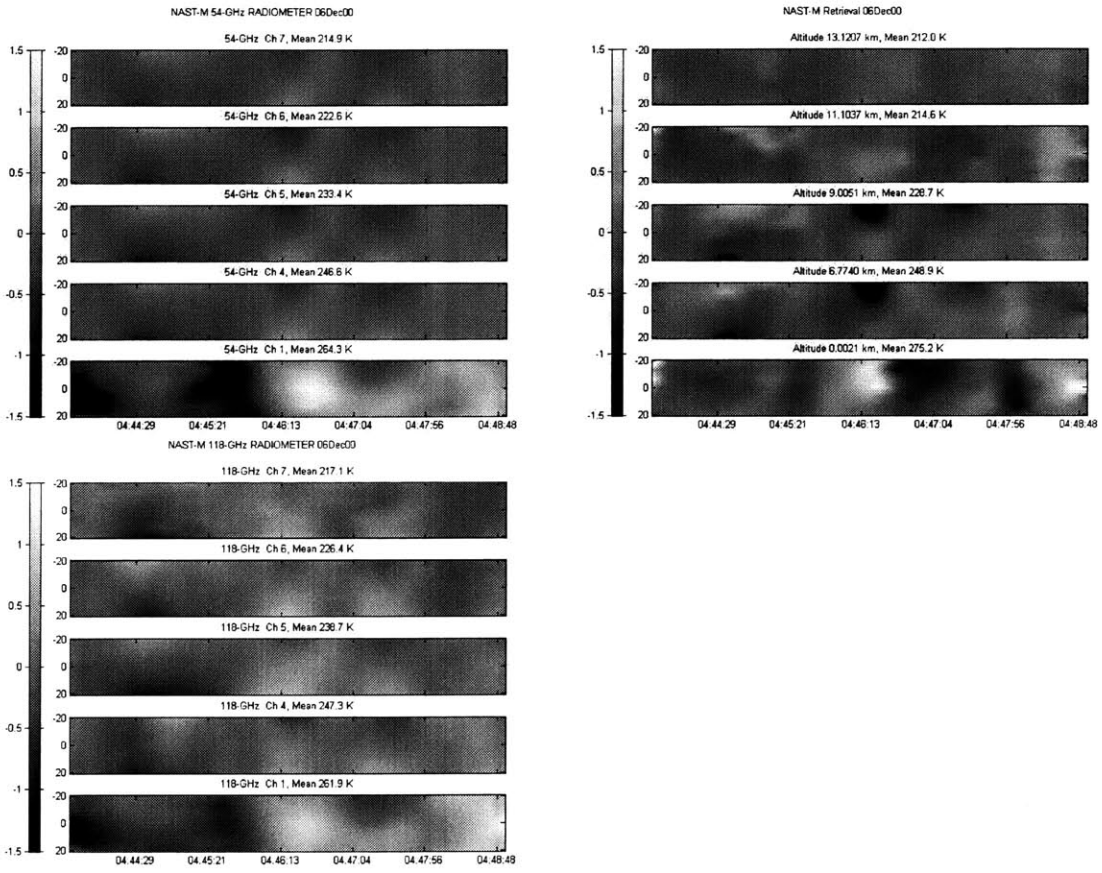


Figure C-80 06Dec00 Temperature Imagery for index 1669.

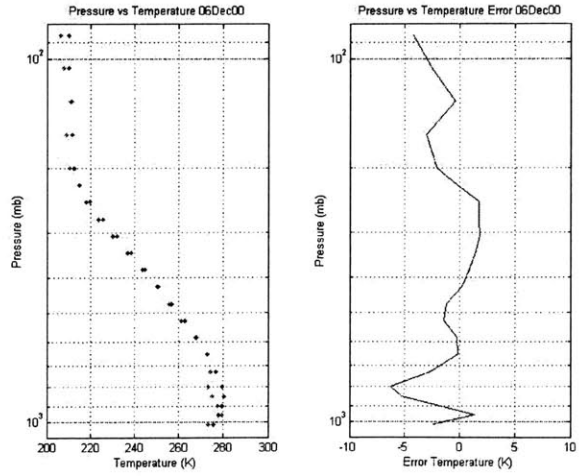


Figure C-81 06Dec00 Temperature Retrieval Comparison for index 1885.

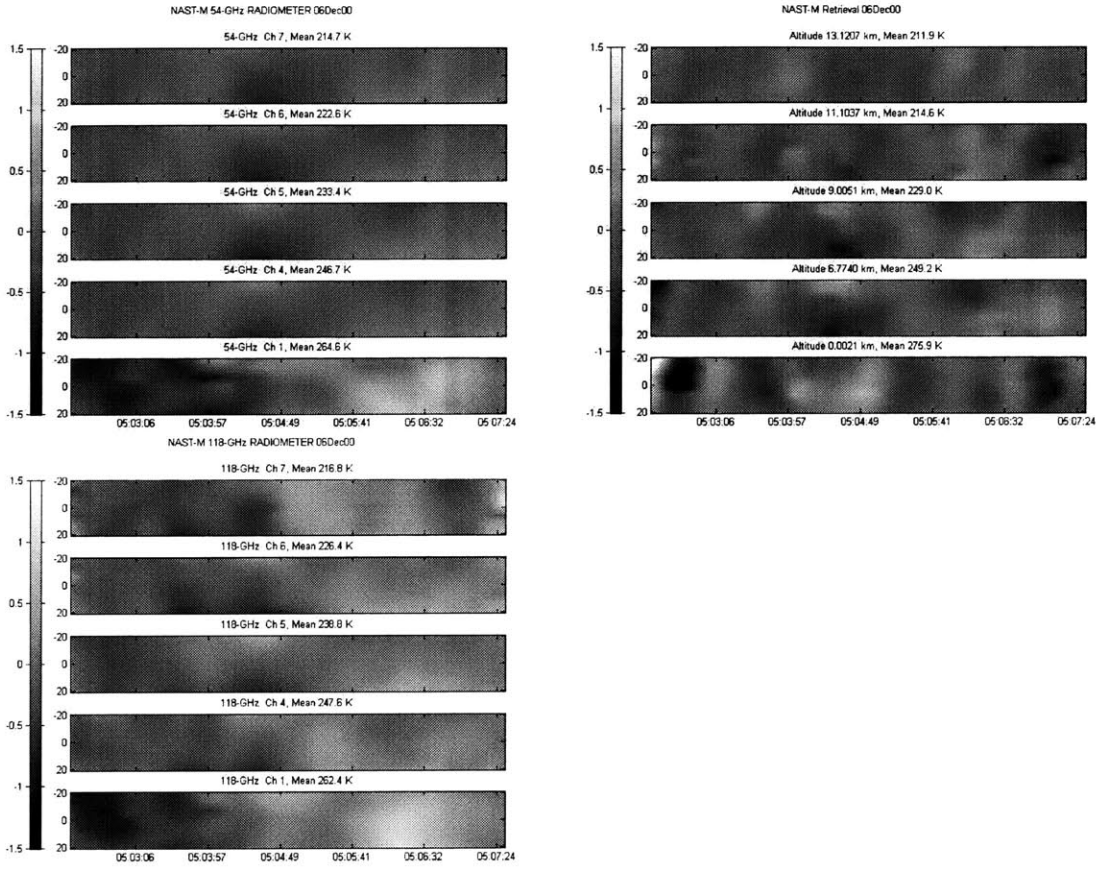


Figure C-82 06Dec00 Temperature Imagery for index 1885.

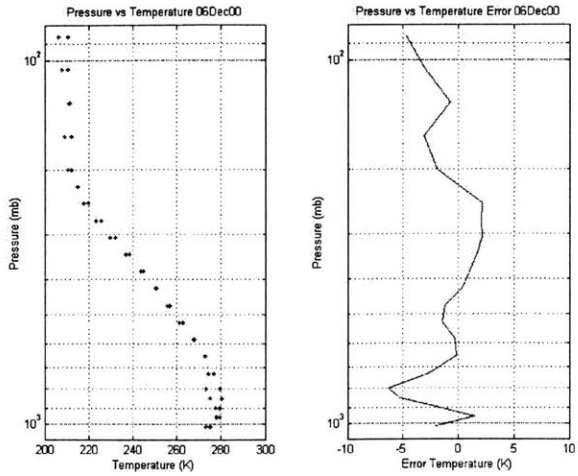


Figure C-83 06Dec00 Temperature Retrieval Comparison for index 1981.

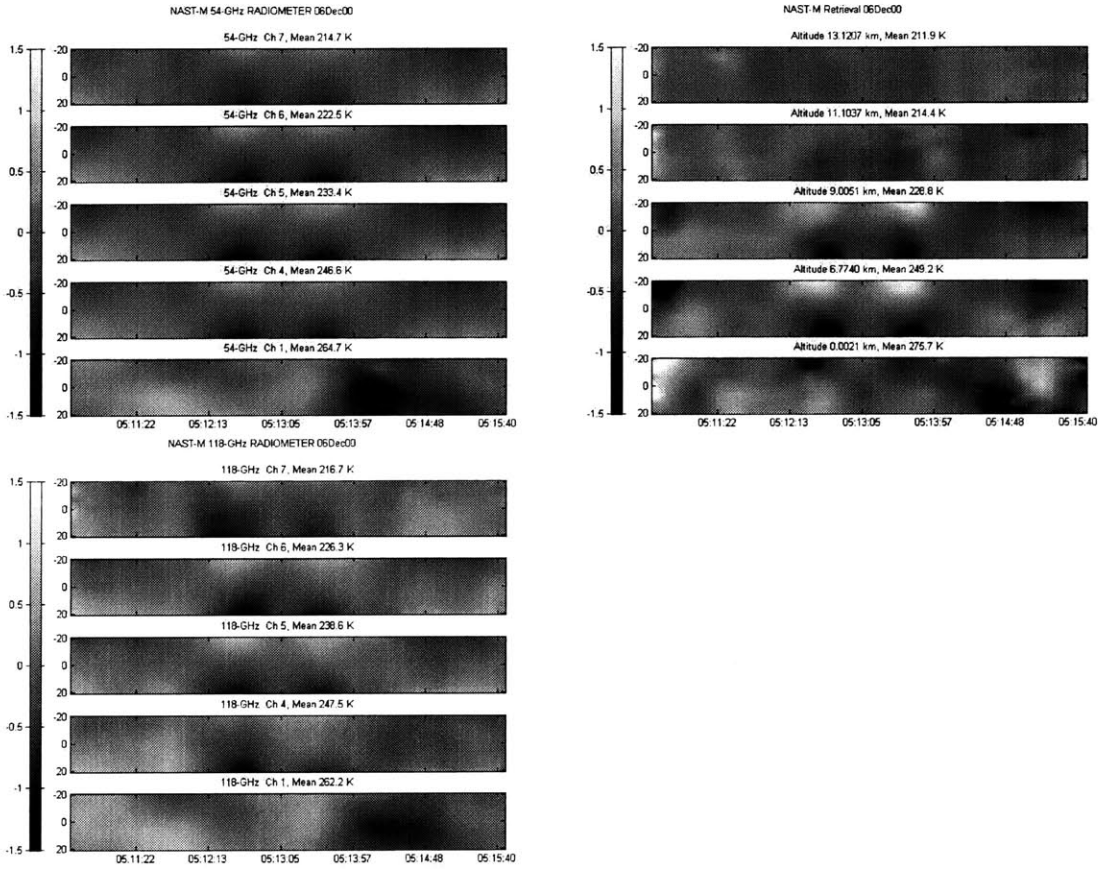


Figure C-84 06Dec00 Temperature Imagery for index 1981.

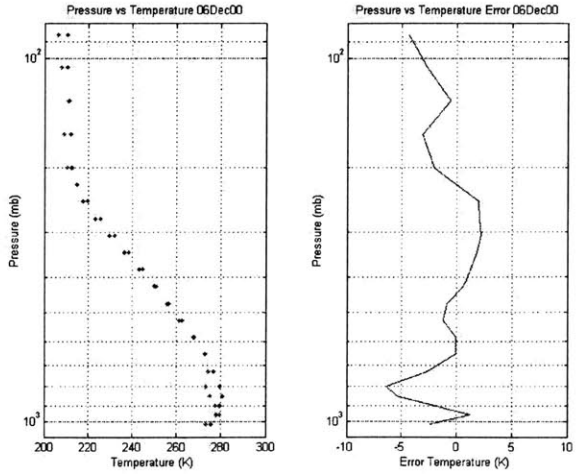


Figure C-85 06Dec00 Temperature Retrieval Comparison for index 2089.

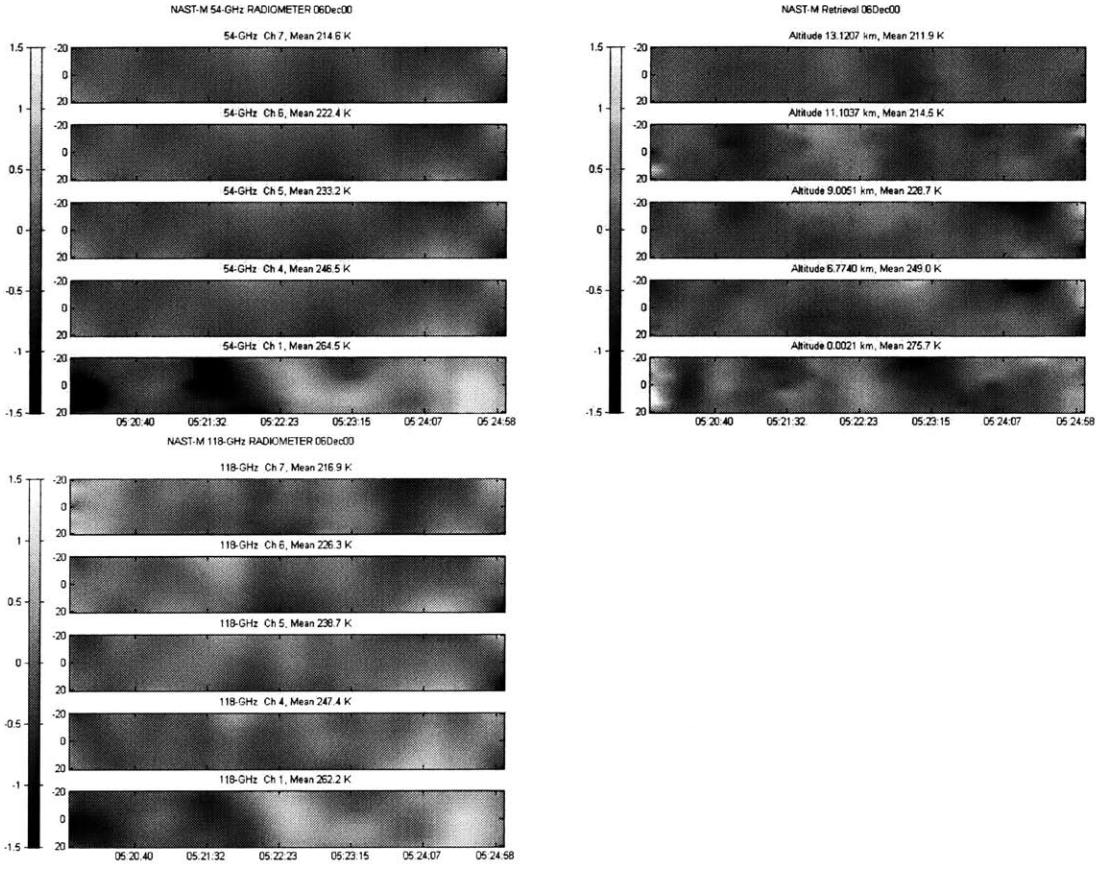


Figure C-86 06Dec00 Temperature Imagery for index 2089.

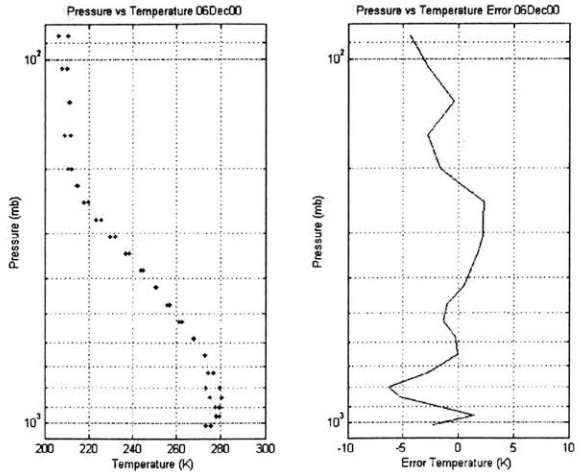


Figure C-87 06Dec00 Temperature Retrieval Comparison for index 2197.

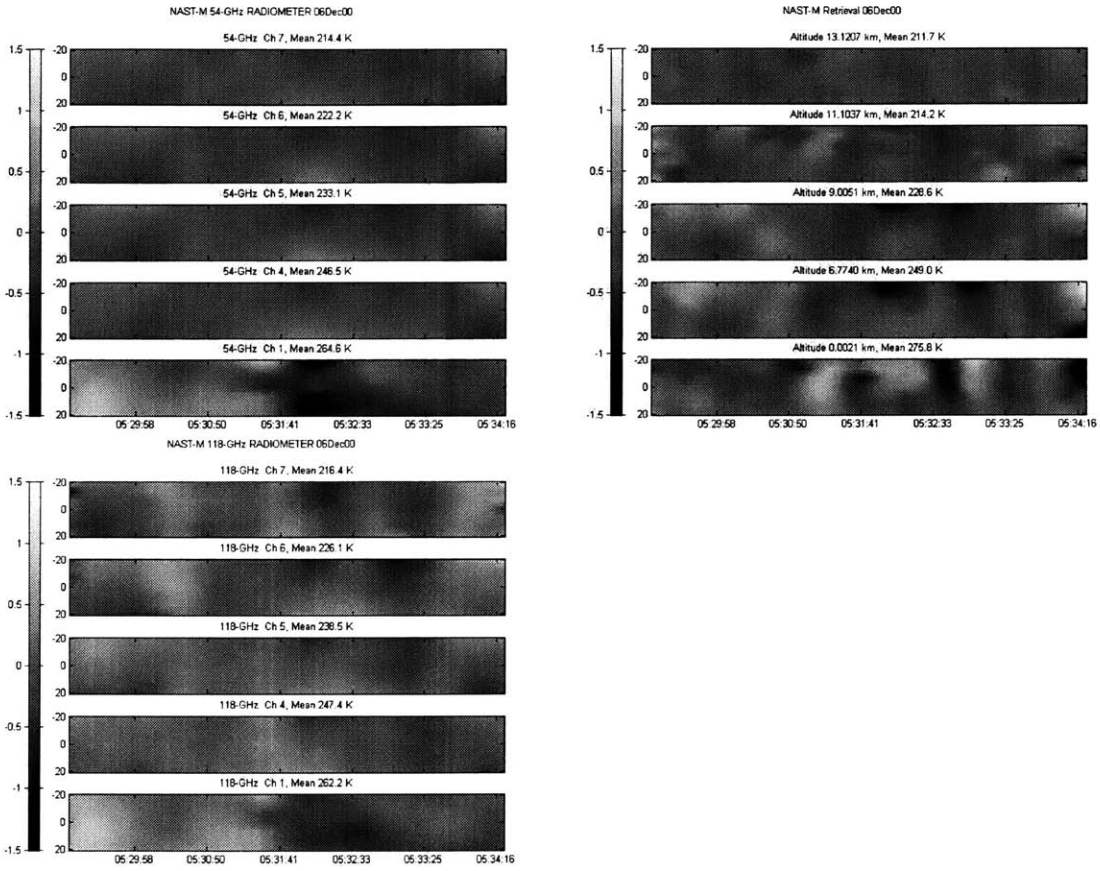


Figure C-88 06Dec00 Temperature Imagery for index 2197.

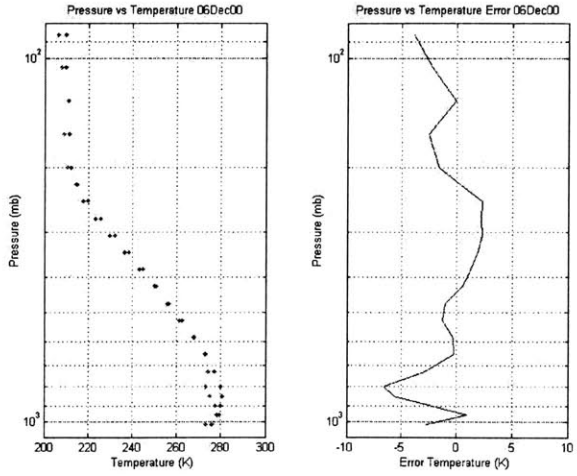


Figure C-89 06Dec00 Temperature Retrieval Comparison for index 2305.

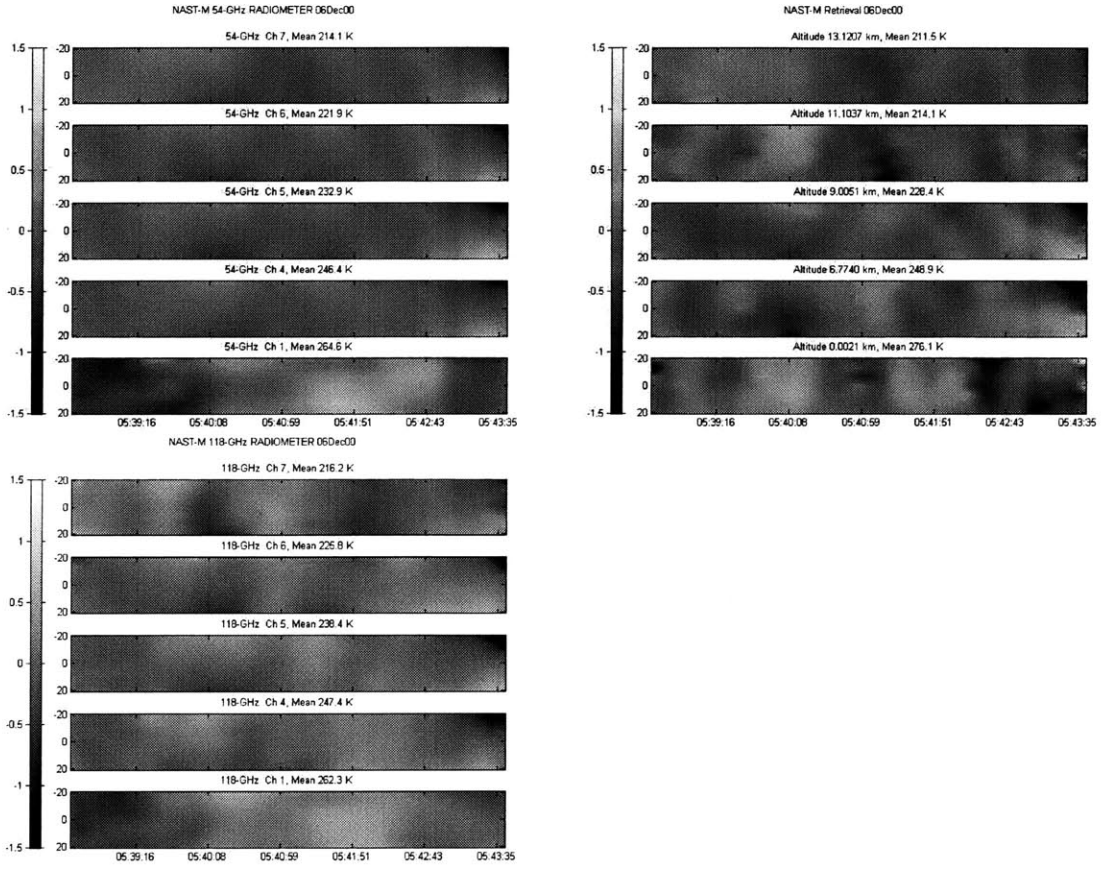


Figure C-90 06Dec00 Temperature Imagery for index 2305.

Bibliography

[1] D. Q. Wark and H. E. Fleming. Indirect measurements of the atmospheric temperature profiles from satellites. *Monthly Weather Review*, 94:351-362, 1966.

[2] D. H. Staelin. Passive remote sensing at microwave wavelengths. *Proceedings of the IEEE*, 57(4):427-439, April 1969.

[3] W. L. Smith, H. M. Woolf, C. M. Hayden, D. Q. Wark, and L. M. McMillin. The TIROS-N operational vertical sounder. Bulletin of the American Meteorological Society, 60(10):1177-1123, October 1979.

[4] D. H. Staelin. Passive microwave sensing of the atmosphere from space. *Microwave Radiometry & Remote Sensing Applications*, pages 151-167, 1989.

[5] J. T. Houghton. *The Physics of Atmospheres*. Cambridge University Press, 1986.

[6] G. L. Stephens. *Remote Sensing of the Lower Atmosphere*. Oxford University Press, New York, New York, 1994.

[7] D. H. Staelin, A. W. Morgenthaler, and J. A. Kong. *Electromagnetic Waves*. Prentice Hall, 1994.

[8] K. N. Liou. *An Introduction to Atmospheric Radiation*. Academic Press, Orlando Florida, 1980.

[9] J. B. Hancock. Passive microwave and hyperspectral retrievals of atmospheric water vapor profiles. Master's thesis, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, May 2001.

[10] S. Chandrasekhar. *Radiative Transfer*. Dover Inc., New York, 1960.

[11] Baxter H. Armstrong and Ralph W. Nicholls. *Emission, Aborption and Transfer of Radiation in Heated Atmospheres*. Pergamon Press, New York, 1972.

[12] P. W. Rosenkranz. Atmospheric Remote Sensing by Microwave Radiometry, chapter Absorption of microwaves by atmospheric gases, pages 37-82. Wiley, 1993.

[13] G. Strang. *Introduction to Linear Algebra*. Wellesley-Cambridge Press, New York, 1998.

[14] G. Strang. *Linear Algebra and its Applications*. Saunders College Publishing, Fort Worth, 1988.

[15] W. J. Blackwell, J. W. Barrett, P. W. Rosenkranz, M. J. Schwartz, and D. H. Staelin. NPOESS Aircraft Sounder Testbed-Microwave (NAST-M): Instrument description and initial flight results. *IEEE International Geoscience and Remote Sensing Symposium Proceedings*, July 2000.

[16] W. J. Blackwell, F. W. Chen, R. V. Leslie, P. W. Rosenkranz, M. J. Schwartz, and D. H. Staelin. NPOESS Aircraft Sounder Testbed-Microwave (NAST-M): Results from CAMEX-3 and WINTEX. IEEE International Geoscience and Remote Sensing Symposium Proceedings, July 2000.

[17] W. J. Blackwell, J. W. Barrett, F. W. Chen, R. V. Leslie, P. W. Rosenkranz, M. J. Schwartz, and D. H. Staelin. NPOESS aircraft sounder testbed-microwave (NAST-M): Instrument description and initial flight results. *IEEE Transactions on Geoscience and Remote Sensing*, 39(11):2444{2453, November 2001.

[18] R. V. Leslie. Temperature profile retrievals with the NAST-M passive microwave spectrometer. Master's thesis, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, June 2000.

[19] W. J. Blackwell. Retrieval of cloud-cleared atmospheric temperature profiles from hyperspectral infrared and microwave observations. ScD thesis, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, Jun 2002.

[20] G. E. Peckham. An optimum calibration procedure for radiometers. *International Journal of Remote Sensing*, 10(1):227-236, 1989.

[21] J. Escobar. *Base de Données pour la Restitution de Paramètres Atmosphériques à L'échelle Globale-Etude sur L'inversion per Réseaux de Neurones des Données des Sondeurs Verticaux Atmosphériques Satellitaires Présents et à venir*. PhD thesis, Ecole Polytechnique, France, 1993.